scientific reports

Check for updates

OPEN

The self supervised multimodal semantic transmission mechanism for complex network environments

Jiajun Zou[1], Zhiping Wan[1], Feng Wang[1], Shitong Ye[2] & Shaojiang Liu[1]✉

With the rapid development of intelligent transportation systems, the challenge of achieving efficient and accurate multimodal traffic data transmission and collaborative processing in complex network environments with bandwidth limitations, signal interference, and high concurrency has become a key issue that needs to be addressed. This paper proposes a Self-supervised Multi-modal and Reinforcement learning-based Traffic data semantic collaboration Transmission mechanism (SMART), aiming to optimize the transmission efficiency and robustness of multimodal data through a combination of self-supervised learning and reinforcement learning. The sending end employs a self-supervised conditional variational autoencoder and Transformer-DRL-based dynamic semantic compression strategy to intelligently filter and transmit the most core semantic information from video, radar, and LiDAR data. The receiving end combines Transformer and graph neural networks for deep decoding and feature fusion of multimodal data, while also using reinforcement learning self-supervised multi-task optimization engine to collaboratively enhance multiple task scenarios (such as traffic accident detection and vehicle behavior recognition). Experimental results show that SMART significantly outperforms traditional methods in low signal-to-noise ratio, high packet loss rate, and large-scale concurrency environments, excelling in key indicators such as semantic similarity, transmission efficiency, robustness, and end-to-end latency, demonstrating its effectiveness and innovation in smart transportation scenarios.

Keywords Intelligent transportation, Multimodal semantic communication, Self-supervised learning, Reinforcement learning, Graph neural network

With the rapid rise of intelligent transportation systems and Vehicle-to-Everything (V2X) technology, vehicles are gradually transforming from traditional means of transportation into intelligent nodes in a connected network. Through the collaboration of sensors, communication modules, and cloud platforms, traffic flow management, road safety, and the driving experience have all seen significant improvements[1–3]. Nevertheless, in the face of ever-increasing urban traffic volumes and diverse driving scenarios, the network environment has become increasingly complex. High-speed vehicle mobility, bandwidth fluctuations, and packet loss occur frequently, posing major challenges for real-time and efficient transmission and collaborative processing of multimodal data in vehicular networks[4–7]. In particular, from the vehicle perception layer, traffic data at any given moment often include heterogeneous forms such as video captured by cameras, radar measurements, and lidar point clouds. How to deeply extract and fuse these data at the semantic level—ensuring transmission stability while contending with limited bandwidth and meeting the timely requirements of key tasks such as traffic incident detection and intelligent dispatch—remains a core challenge urgently awaiting breakthrough[8–10]. In existing C-V2X commercial deployments, the downlink/uplink effective bandwidth is typically only 5–20 MHz, multipath fading causes SNR fluctuations from −3 to 18 dB at high speeds (>100 km h^{-1}), bursty interference and cellular switching can push instantaneous packet loss rates up to 10–30%, and end-to-end delay budgets still need to be kept within 100 ms for collision warning. The end-to-end delay budget still needs to be kept within 100 ms for collision warning. These quantitative metrics highlight the triple paradox of high bandwidth overhead, unstable channels, and demanding real-time performance, which makes multimodal semantic communication substantially more difficult.

In traditional solutions, multimodal data are often processed offline or in near real-time at the terminal or cloud side, for example, through video image recognition and radar-based target detection. However,

[1]School of Information and Intelligence Engineering, Guangzhou Xinhua University, Dongguan 523133, China. [2]Artificial Intelligence Institute of Guangzhou Huashang College, Guangzhou 511300, China. ✉email: mrliuxinhua@xhsysu.edu.cn

under poor network conditions or high vehicle concurrency, the direct transmission of large-scale raw data can severely occupy bandwidth and cause congestion or packet loss, thus affecting the overall effectiveness of intelligent transportation systems. Semantic communication[11–13] provides a potential approach: by precisely compressing and transmitting the "core meaning" of the data rather than the raw bitstream, it is possible to minimize redundancy and emphasize retention of useful information in noisy environments. Yet, most existing semantic communication research focuses on single-modal scenarios (text or images) or primarily addresses lower-level power and spectrum allocation in vehicular networks. Research on the synergy of multimodal semantic extraction, dynamic transmission, and deep decoding at the receiver remains insufficient[11,14–16]. Moreover, intelligent transportation systems impose stringent requirements for real-time performance and stability. Under high-speed mobility, occlusions, nighttime conditions, or extreme weather, continuous and accurate semantic perception and anomaly detection are needed for vehicles and their surroundings[17–19]. As smart cities continue to expand, the complex communication network formed among roadside units, cloud platforms, and massive vehicles makes bandwidth allocation and packet loss a regular concern. Relying on conventional routing or physical-layer optimizations alone falls short of addressing the dynamic compression needs of upper-layer multimodal data, thereby failing to fundamentally improve the transmission quality of multimodal traffic data under varying network conditions. Effectively screening and compressing video, radar, and lidar data at the sender side—and employing adaptive decoding and multitask learning approaches at the receiver side—are pressing issues researchers are eager to resolve.

Based on this, this paper proposes a self-supervised multimodal traffic data semantic collaborative transmission mechanism for complex network environments with respect to the efficient transmission and semantic parsing of large-scale multimodal traffic data. On the sender side, a self-supervised conditional variational autoencoder (CVAE) is used to perform deep semantic extraction on multi-source sensing data such as video, radar, and lidar. A Transformer-DRL-based dynamic compression scheme adaptively regulates the transmission strategy, enhancing robustness and bandwidth utilization. On the receiver side, a combination of a Transformer decoder and graph neural networks is employed for multimodal fusion. A self-supervised multitask reinforcement learning engine further optimizes tasks such as accident detection and behavior recognition in a collaborative manner, ensuring that crucial multimodal information can still be accurately reconstructed in extreme channel conditions. The objective of this study is not only to validate the efficiency and stability of multimodal data under a semantic communication framework but also to provide a feasible communications-perception convergence paradigm for future intelligent transportation systems.

The main contributions of this paper are as follows:

1. Multimodal Self-Supervised Semantic Extraction and Adaptive Compression: A collaborative mechanism of a self-supervised conditional variational autoencoder (CVAE) and a Transformer-DRL framework is proposed, enabling semantic-level selection and dynamic compression of multi-source heterogeneous data from video, radar, and lidar point clouds at the sender side. This fully utilizes network bandwidth and enhances robustness under low signal-to-noise ratio environments.
2. Receiver-Side Multimodal Fusion and Multitask Reinforcement Learning: By leveraging Transformer-GNN for in-depth multimodal feature integration and a self-supervised multitask reinforcement learning framework for accident detection and behavior recognition, the system ensures that core traffic semantics are retained even under high packet loss rates or varying communication delays.
3. Comprehensive Experiments and Comparisons: Extensive experiments are conducted to systematically compare the performance of SMART, DeepSC, and SSS in terms of semantic similarity, transmission efficiency, robustness, and latency. In simulated vehicular network contexts, SMART demonstrates superior performance under complex network conditions.

The remainder of this paper is organized as follows: Section "Introduction" reviews the related literature on vehicular network semantic communications and multimodal semantic communications. Section "Related work" introduces the overall approach and system architecture of SMART. Section "System modeling" describes the sender-side self-supervised CVAE semantic extraction and Transformer-DRL dynamic compression mechanism in detail. Section "Transmitter" explains the receiver-side multimodal fusion and the multitask optimization process under self-supervised reinforcement learning. Section "Experimental analysis" presents and analyzes the experimental design and comparison results. Finally, Section "Conclusion" summarizes the paper and discusses future research directions.

Related work

Semantic communication in vehicular networks

In recent years, semantic communication concepts have been gradually applied to connected vehicles (C-V2X) and intelligent transportation systems to improve communication efficiency and task completion. Combining reinforcement learning with semantic awareness has become a major research trend in Telematics scenarios. For example, Shao et al. proposed a semantically aware resource management mechanism for fleet formation, which utilizes multi-intelligence reinforcement learning (MARL) to optimize the communication resource allocation, including spectrum channel selection, transmission power, and semantic symbol lengths, based on the semantic importance of the transmitted information, so as to in the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications to prioritize the data that is critical for the safe and efficient operation of the fleet. However, this approach is mainly optimized for specific fleet scenarios, and the limitation is that the proposed semantic resource management, although considering multi-task multimodal data, still has limited support for compressed transmission of truly complex multimodal information[20]. Shao et al. further investigated the semantic spectrum sharing problem in high-speed mobile IoT vehicular environments by proposing the SSS

algorithm, which employs the Soft Actor-Critic Algorithm (SAC) in Deep Reinforcement Learning to realize semantics-aware spectrum sharing decisions. The method first extracts semantic information in vehicular communication, and then redefines the semantic information evaluation metrics in spectrum sharing scenarios, such as High Speed Semantic Spectrum Efficiency (HSSE) and Semantic Transmission Rate (HSR), in order to portray the effectiveness of semantic data transmission in V2V and V2I concurrent communication[21]. However, this work focuses on resource optimization in the spectrum dimension, and defaults to a consistent understanding of semantic information at the sender and receiver ends, but still lacks in-depth discussion on how to efficiently compress multimodal semantic content and how to timely adjust the strategy when the network is rapidly changing. Vanneste et al. investigated methods for learning communication in adaptive traffic control in their scenario consisting of connected intelligent traffic signals, each intersection controlled by independent intelligences that can observe only local traffic states. The work compares independent Q-learning without communication with a multi-intelligent reinforcement learning approach that introduces a differentiable communication mechanism. However, the communication content in this method is implicitly learned by the algorithm and lacks explicit modeling of the semantic meaning of the message, making it difficult to ensure the interpretability of the transmitted information to humans or external systems[22].

Multimodal semantic communication

Facing emerging application scenarios such as 6G and meta-universe, communication systems need to transmit semantic information containing multiple modalities such as image, audio, text, depth, etc., and the research on multimodal semantic communication is thus rapidly developing. Early semantic communication research such as DeepSC system proposed by Xie et al. utilizes deep learning to extract sentence semantics end-to-end for encoding and transmission (based on the Transformer model), which significantly reduces transmission bits with the optimization goal of reconstructing semantic accuracy on the text transmission task[23]. To support cross-modal semantic alignment and communication, Li et al. proposed a framework conceptualization of cross-modal semantic communication, emphasizing that all participating nodes should share the same or similar knowledge base (e.g., public knowledge graph) to understand the same semantic concepts expressed in different modalities[24]. Chen et al. proposed a cross-modal graph semantic communication method for meta-universe 6G scenarios. The method utilizes GNN to extract key semantic features of multimodal data (e.g., images and 3D point clouds) and compressed representation of these semantic features at the sender's end by a graph Transformer encoder, which employs a cross-modal attention mechanism to fuse information from different modalities[25]. However, this scheme is optimized for the fusion and reconstruction of specific types of modalities (image and point cloud) and is not validated for other types of data (e.g., text) or other tasks. Zhao et al. propose a multimodal self-supervised semantic communication framework that aims to reduce the communication cost of model training and updating in dynamic wireless environments. The method employs two-stage training: first multimodal self-supervised pre-training to learn task-independent modal invariant representations, and then supervised fine-tuning for specific downstream tasks. However, the method does not provide an in-depth discussion on how rapid changes in the wireless channel (e.g., bursty interference, bandwidth fluctuations) affect semantic encoding and decoding in real deployments, and thus needs to be further strengthened in terms of dynamic network state awareness[26].

System modeling

In this study, a multimodal traffic data semantic collaborative transmission mechanism driven by reinforcement learning and self-supervision mechanism is proposed to address the problem of efficient transmission and understanding of multimodal traffic data semantic information in intelligent transportation systems under complex network environments. The overall system model is mainly composed of two core modules, the sender side and the receiver side, as shown in Fig. 1.

The transmitter collects multimodal traffic scene data in real time through front-end sensing devices, including video sequences, radar data and point cloud data. These heterogeneous data are individually feature extracted by specially designed feature encoding networks, in which the visual data capture spatial and temporal semantic features using a hybrid CNN-Transformer network, the radar data encode temporal characteristics through bi-directionally gated cyclic units, and the geometric spatial semantics are extracted from LIDAR point cloud data using a PointNet structure. Subsequently, the multimodal data features are adaptively fused by the proposed multi-head cross-modal attention mechanism to form a unified, compact and representative semantic feature vector.

Based on the multimodal feature fusion, the sender further proposes a self-supervised Conditional Variational Autoencoder (CVAE) model[27–29] for low-dimensional semantic compression of the fused features. The conditional encoder generates conditional probability distributions of semantic latent variables guided by self-supervised signals, and the conditional decoder is used to reconstruct the fused features to optimize model training by maximizing the lower bound of variational evidence. And the contrast learning loss function is innovatively introduced to enhance the discriminative and generalization ability of the semantic latent variables to obtain a low-dimensional and robust semantic latent variable representation.

Considering that real complex network environments are usually characterized by bandwidth constraints, latency fluctuations, and uncertain packet loss rates, this study further designs a dynamic semantic compression transmission mechanism based on the fusion of Transformer[30–32] and Deep Reinforcement Learning (DRL)[33–36] at the sender side. Transformer-based Semantic Compression Module (TSCM) dynamically mines semantic correlations between latent variable dimensions through a multi-head self-attention mechanism to achieve semantic context enhancement and accurate compression. The Semantic Compression Control Policy Network (SCCPN) driven by reinforcement learning monitors the network status in real time and dynamically adjusts the feature compression factor to ensure the transmission efficiency and stability of semantic data. The whole

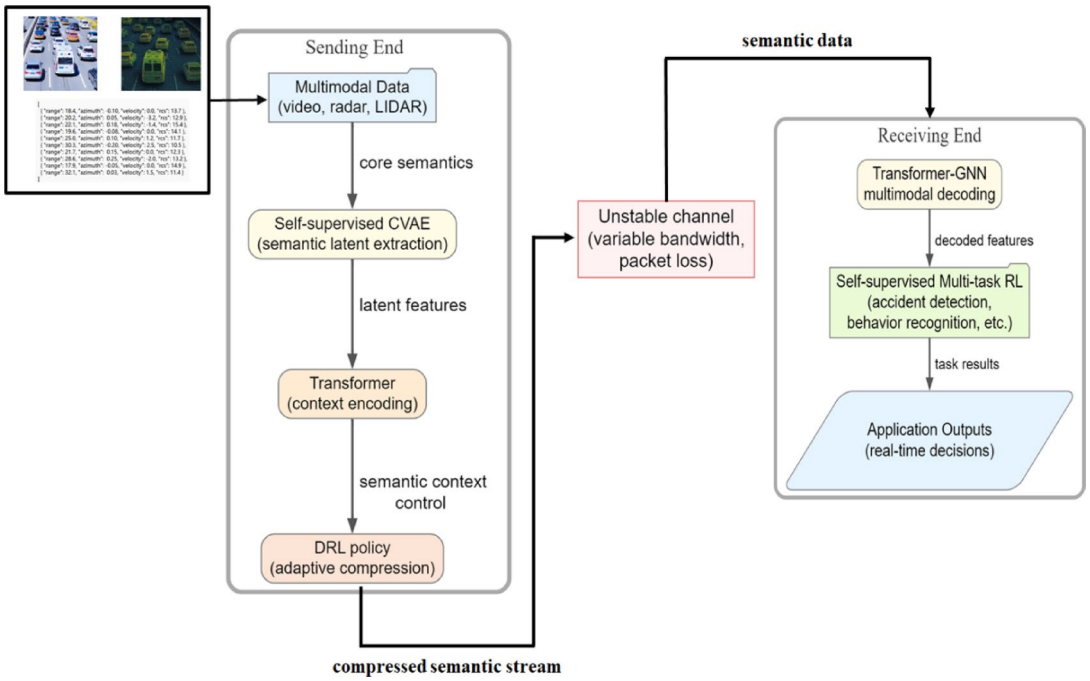

Fig. 1. System model.

semantic compression and transmission control process is modeled as Markov Decision Process (MDP), and the Actor-Critic framework and experience playback mechanism are adopted to efficiently realize the adaptive compression and transmission of semantic data.

The receiver receives the semantic latent variable sequences transmitted by the Transformer-DRL dynamic compression, and firstly employs the Transformer decoder to recover and enhance the semantic context features. The recovered semantic features are fused with the local multimodal traffic environment information, and the preliminary semantic feature enhancement is realized through the cross-modal attention mechanism. Subsequently, a Spatial Semantic Graph (SSG) is constructed, and the spatial semantic interactions between different traffic entities are further modeled by Graph Neural Network (GNN), and the deep propagation and fusion of node features are achieved by using multi-layer graph convolutional network and multi-modal graph attention mechanism to obtain a more fine-grained and comprehensive semantic representation. The deep propagation and fusion of node features are realized by using multi-layer graph convolutional network and multi-head graph attention mechanism to obtain a more refined and comprehensive semantic expression.

Considering that the practical applications of intelligent transportation systems often need to perform multiple complex tasks (e.g., accident detection, behavior recognition, scene understanding) simultaneously, and there are semantic correlations and potential conflicts among these tasks, this study further proposes a Reinforcement Learning-based Self-supervised Multi-task Optimization Engine (RL-SMOE). Multi-task Optimization Engine (RL-SMOE). The engine dynamically adjusts the weight allocation of the multi-task network to collaboratively optimize the task performance through a deep deterministic policy gradient algorithm (DDPG). The innovative introduction of self-supervised perturbation signals provides stability-oriented additional rewards for reinforcement learning strategies, which significantly improves the generalization ability of the strategy network in dynamic environments.

Transmitter

In the multimodal data transmission process of intelligent transportation system, the sender not only needs to extract preliminary semantic information for heterogeneous data such as video, radar and LIDAR, but also needs to consider the fluctuating characteristics of the network environment in order to realize dynamic compression and robust transmission of high-dimensional semantic features. In this study, the CVAE model is introduced at the sender side to extract uniform and compact semantic latent variables from multimodal traffic scenarios. By combining the dynamic semantic compression and transmission mechanism of Transformer and DRL, we adaptively balance the network bandwidth, delay and data loss to ensure the accurate delivery of key semantic information and real-time interaction. The sender framework (shown in Fig. 2) takes into account multimodal fusion and network adaptivity, aiming to address the multidimensional challenges of semantic information transmission in the intelligent transportation environment.

Self-supervised conditional variational self-encoder semantic extraction

The first problem faced by the sender side of this study is how to effectively extract unified and compact semantic information from multimodal data (including video sequences, radar data, and LIDAR LIDAR data) of complex

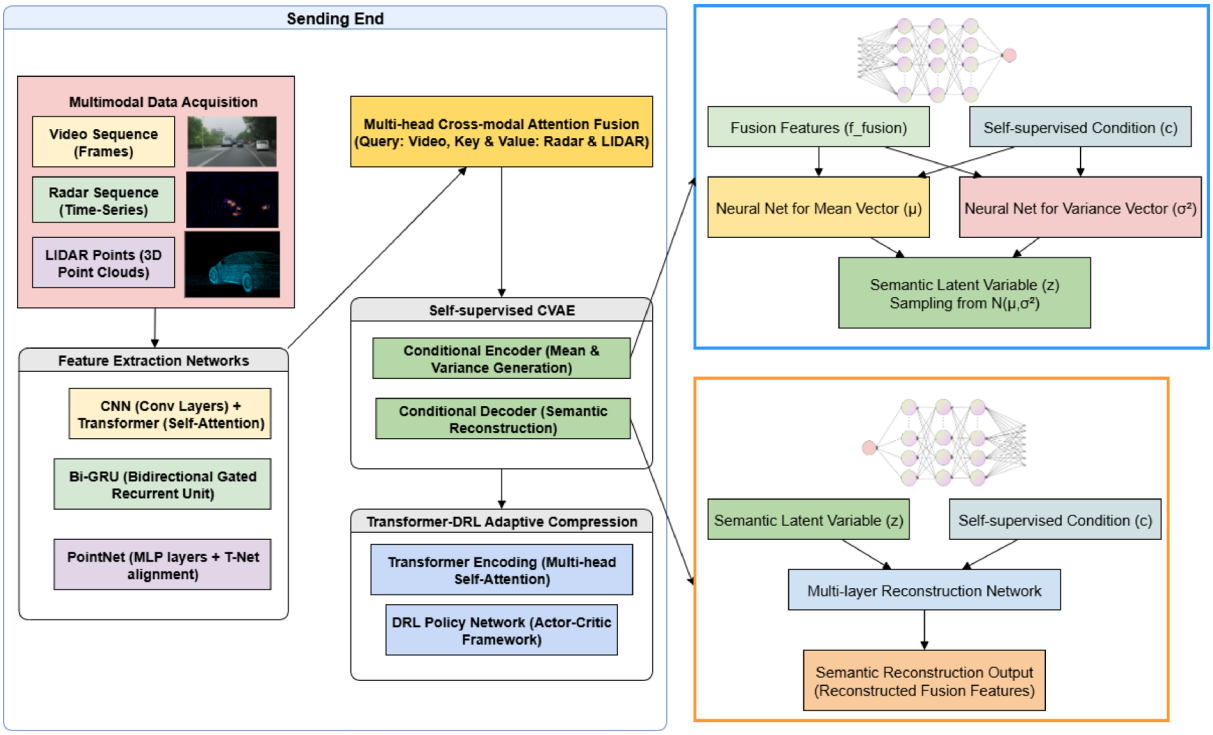

Fig. 2. Transmitter framework.

intelligent transportation scenarios for efficient transmission in complex network environments. In this section, we propose a self-supervised conditional variational self-encoder (method for achieving automatic semantic information extraction from multimodal traffic data.

The system collects multimodal data in the traffic scene in real time from the front-end sensing devices, including video data $\mathbf{V} = \{V_t\}_{t=1}^{T}$ (each frame of the image V_t has the dimension $H \times W \times 3$), radar sequence data $\mathbf{R} = \{R_t\}_{t=1}^{T}$ (with the dimension D_R), and LIDAR point cloud sequence data $\mathbf{L} = \{L_t\}_{t=1}^{T}$ (each moment consists of N_p 3D points). In order to process these heterogeneous data efficiently, we design feature coding networks for different modalities separately to initially extract the semantic information of the data modalities themselves. The video sequences use a hybrid CNN-Transformer network structure, where the CNN network is responsible for capturing the in-frame spatial features, while the Transformer network is used to capture the long-range semantic dependencies across time frames to obtain the visual feature representation $\mathbf{f}_V \in \mathbb{R}^{d_v}$. The radar sequence data, on the other hand, captures its temporal dynamic properties through a bidirectional gated loop unit to obtain the radar feature representation $\mathbf{f}_R \in \mathbb{R}^{d_r}$.The LIDAR data uses a PointNet-based structure to extract geometric and spatial structural features to form the LIDAR feature representation $\mathbf{f}_L \in \mathbb{R}^{d_l}$).

We use the visual modal feature $\mathbf{f}_V$ as the Query vector, while the radar feature $\mathbf{f}_R$ is spliced with the LIDAR feature $\mathbf{f}_L$ as the Key and Value vectors (notated as $\mathbf{F}_{RL} = [\mathbf{f}_R; \mathbf{f}_L)$. The fusion feature vector $\mathbf{f}_{fusion}$ is obtained by computing the multi-head cross-modal attention mechanism, and the fusion mechanism is specifically expressed as:

$$\mathbf{f}_{fusion} = \text{Concat}(\text{head}_1, ..., \text{head}_h)\mathbf{W}^O \tag{1}$$

where each attention head is computed as:

$$\text{head}_i = \text{softmax}(\frac{(\mathbf{f}_V \mathbf{W}_i^Q)(\mathbf{F}_{RL}\mathbf{W}_i^K)^T}{\sqrt{d_k}})\mathbf{F}_{RL}\mathbf{W}_i^V \tag{2}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ and $\mathbf{W}^O$ are the projection matrices for network training, and d_k denotes the dimension of the attention vector in each head. In this paper, the cross-modal attention fusion method enhances the synergy and stability of inter-modal semantic features by explicitly modeling the semantic associations between different modal data and dynamically assigning different weights to each modal feature.

After completing the multimodal data feature fusion, we obtain a unified semantic feature expression $\mathbf{f}_{fusion}$, however, this fused high-dimensional feature is not suitable for direct transmission in complex network environments. Therefore, we introduce the CVAE model to further realize the semantic compression and self-supervised learning of the fused features. Our CVAE model contains two core modules, the conditional encoder and the conditional decoder. In the conditional encoder, the fused feature $\mathbf{f}_{fusion}$ and the self-supervised condition $\mathbf{c}$ are jointly used as inputs to generate the mean vector $\boldsymbol{\mu}_\phi$ and the variance vector $\boldsymbol{\sigma}_\phi^2$ of the latent

semantic variable $\mathbf{z}$ through two independent deep neural networks, respectively. The distribution form of the conditional encoder is defined as a conditional Gaussian distribution:

$$q_\phi(\mathbf{z}|\mathbf{f}_{fusion}, \mathbf{c}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{f}_{fusion}, \mathbf{c}), \boldsymbol{\sigma}^2_\phi(\mathbf{f}_{fusion}, \mathbf{c})\mathbf{I}) \tag{3}$$

Condition $\mathbf{c}$ comes from the self-supervised signal obtained from random data enhancement (e.g., local masking, feature perturbation) performed on the fusion feature itself. The conditional decoder reconstructs the original fusion feature vector $\mathbf{f}_{fusion}$ using the semantic latent variable $\mathbf{z}$ and the self-supervised condition $\mathbf{c}$ as inputs. the specific decoding process is expressed as:

$$p_\theta(\mathbf{f}_{fusion}|\mathbf{z}, \mathbf{c}) = \mathcal{N}(\mathbf{f}_{fusion}; \boldsymbol{\mu}_\theta(\mathbf{z}, \mathbf{c}), \boldsymbol{\sigma}^2_\theta(\mathbf{z}, \mathbf{c})\mathbf{I}) \tag{4}$$

To train this model, we use a variational inference framework with the goal of maximizing the Evidence Lower Bound (ELBO) with the loss function defined as follows:

$$\mathcal{L}_{CVAE}(\phi, \theta) = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{f}_{fusion}, \mathbf{c})}[\log p_\theta(\mathbf{f}_{fusion}|\mathbf{z}, \mathbf{c})] + \beta \cdot KL(q_\phi(\mathbf{z}|\mathbf{f}_{fusion}, \mathbf{c}) \parallel p_\theta(\mathbf{z}|\mathbf{c})) \tag{5}$$

where the first term (reconstruction loss) is used to ensure that the semantic information carried by the latent variable $\mathbf{z}$ can be effectively restored to the original fusion features, thus avoiding excessive loss of semantic information, while the second KL scattering term serves as a means of regularization to make the latent variable distribution close to the semantic prior distribution given the condition $\mathbf{c}$:

$$p_\theta(\mathbf{z}|\mathbf{c}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{prior}_\theta(\mathbf{c}), \boldsymbol{\sigma}^{prior2}_\theta(\mathbf{c})\mathbf{I}) \tag{6}$$

where β is a weighting hyperparameter used to trade-off between the reconstruction loss and the KL scatter loss, which can be dynamically adjusted according to the complexity of the network environment and the transmission demand to realize the optimal compression ratio of the latent variable representation.

To further enhance the robustness and discriminative ability of semantic feature latent variables, we innovatively introduce a comparative learning loss function in the latent variable space, which is used to further constrain the distribution of latent variables in the semantic space. We define a latent variable-based contrast loss as follows:

$$\mathcal{L}_{contrastive}(\mathbf{z}, \mathbf{z}^+, \mathbf{z}^-) = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}, \mathbf{z}^+)/\tau)}{\exp(\mathrm{sim}(\mathbf{z}, \mathbf{z}^+)/\tau) + \sum_j \exp(\mathrm{sim}(\mathbf{z}, \mathbf{z}^-_j)/\tau)} \tag{7}$$

where $\mathbf{z}^+$ is the latent variable corresponding to augmented data from the same semantic category, $\mathbf{z}^-$ is the latent variable from different semantic categories or different data samples, $\mathrm{sim}(\cdot)$ denotes the cosine similarity function, and τ is the temperature parameter. By jointly optimizing $\mathcal{L}_{CVAE}$ and $\mathcal{L}_{contrastive}$, the distinguishability and generalization ability of the semantic representation is improved.

Transformer-DRL-based dynamic semantic compression transmission mechanism

In the semantic extraction stage, we successfully compress the multimodal data of intelligent transportation scenarios into a unified and compact semantic latent variable representation $\mathbf{z}$. However, in real intelligent transportation systems, the network environment usually has complex and variable characteristics, such as bandwidth fluctuation, latency jitter, and packet loss, which will directly affect the real-time performance and reliability of semantic data transmission. Therefore, we need to further perform efficient dynamic semantic compression for the latent variable $\mathbf{z}$ to adapt to the real-time changes of the complex network environment and to ensure the stability and robustness of the data semantics in the transmission process. We propose a dynamic semantic compression transmission mechanism based on the fusion of Transformer and DRL.

Considering that although the latent variable $\mathbf{z}$ generated by CVAE has been compressed to lower dimensions, there is variability in the contribution of semantic information among different dimensions, direct transmission may still cause redundancy and transmission bottlenecks. Therefore, we use TSCM to further explore the semantic correlation between the dimensions of the latent variable, so as to precisely control the degree of compression and reduce the transmission of redundant information. Assuming that the semantic latent variable sequence obtained from CVAE is $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T]$, where each $\mathbf{z}_t \in \mathbb{R}^{d_z}$ represents a semantic latent variable feature at a certain moment in the time series, TSCM firstly projects the feature sequence into the Transformer's feature space by linear mapping:

$$\mathbf{Z}\prime = [\mathbf{z}_1\prime, \mathbf{z}_2\prime, ..., \mathbf{z}_T\prime] = \mathbf{Z}\mathbf{W}_{in} + \mathbf{b}_{in}, \mathbf{W}_{in} \in \mathbb{R}^{d_z \times d_m} \tag{8}$$

In Transformer Encoder, the long range interactions and correlations between the dimensions of the semantic sequence are captured by the self-attention mechanism and the semantic context-enhanced feature representation is generated. The multi-head self-attention computation process is represented as follows:

$$\mathrm{MHSA}(\mathbf{Z}\prime) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)\mathbf{W}^O \tag{9}$$

where each attention head is calculated as:其中每个注意力头的计算方式为:

$$\text{head}_i = \text{softmax}(\frac{(\mathbf{Z}\prime\mathbf{W}_i^Q)(\mathbf{Z}\prime\mathbf{W}_i^K)^T}{\sqrt{d_k}})\mathbf{Z}\prime\mathbf{W}_i^V \tag{10}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V, \mathbf{W}^O$ are trainable weight matrix parameters, d_k denotes the dimension of each attention head, and h is the number of attention heads. After processing by the Transformer Encoder module, we obtain an intermediate representation $\mathbf{Z}^{enc}$ that is contextually semantically enhanced and suitable for compressed transmission.

However, only utilizing the Transformer Encoder for compression representation cannot adapt to the real-time changes of the complex network environment, for this reason, we further incorporate the DRL approach to adaptively adjust the compression degree and transmission strategy of the Transformer module, which we call Semantic Compression Control Policy Network (SCCPN) based on reinforcement learning driven. Compression Control Policy Network (SCCPN). The network takes the state vector $\mathbf{s}_t = [b_t, l_t, p_t]$ consisting of the current network states (bandwidth b_t,delay l_t,packet loss p_t) as input, and outputs the action vector $\mathbf{a}_t$, which controls the compression factor γ_t of the Transformer compression module, and the feature compression is realized as:

$$\gamma_t = \sigma(\mathbf{W}_{act} \cdot \mathbf{a}_t + \mathbf{b}_{act}) \tag{11}$$

$$\mathbf{Z}_t^{comp} = \mathbf{Z}_t^{enc} \odot \gamma_t \tag{12}$$

where $\gamma_t \in (0,1)$, determines the compression ratio of the feature dimension, $\odot$ is Hadamard's element-by-element product operation. To achieve adaptive optimization of semantic data transmission in complex network environments, we next define the practice process of semantic compression strategies. In the previous section, we define SCCPN with real-time network state $\mathbf{s}_t = [b_t, l_t, p_t]$ as input and output action vector $\mathbf{a}_t$ to control the compression factor γ_t of the Transformer compression module.However, in order to train this strategy network effectively, we have to reasonably define the states, actions, rewards, and the training method in reinforcement learning. We model the entire semantic transfer process as a Markov Decision Process (MDP), which is formally defined as a quaternion $(\mathcal{S}, \mathcal{A}, P, R)$, where S denotes the state space (the network state and the semantic data state), $\mathcal{A}$ denotes the action space (the modulation of the degree of feature compression), $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transfer probability function, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. In our approach, the state $\mathbf{s}_t$ is explicitly defined as a vector consisting of the network's current bandwidth, latency, and packet loss rate, the action $\mathbf{a}_t$ is the control of the degree of semantic compression, and the state transfer is reflected in the network environment over time, while the reward function takes into account the semantic compression rate, the transmission latency, and the semantic transmission accuracy, which is defined as:

$$R_t(\mathbf{s}_t, \mathbf{a}_t) = \alpha_1 \frac{d_{ori} - d_{comp}}{d_{ori}} - \alpha_2 l_t - \alpha_3 p_t \tag{13}$$

where d_{ori} is the amount of original semantic feature data, d_{comp} is the amount of compressed data, l_t is the network latency, p_t is the packet loss rate, and $\alpha_1, \alpha_2, \alpha_3$ are the weighting coefficients to balance the compression effect, real-time and reliability. The reward function is designed to ensure that the reinforcement learning process can simultaneously balance transmission efficiency, semantic integrity and network quality. We adopt the Actor-Critic framework in order to realize strategy learning in continuous action space. The strategy network generates actions, the evaluation network assesses strategy performance, and the loss function of the Critic network is defined as the Bellman error:

$$\mathcal{L}(\omega) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}) \sim D}[(r_t + \gamma Q_{\omega\prime}(\mathbf{s}_{t+1}, \pi_{\psi\prime}(\mathbf{s}_{t+1})) - Q_\omega(\mathbf{s}_t, \mathbf{a}_t))^2] \tag{14}$$

where D is the empirical playback buffer pool, γ is the discount factor $(0 < \gamma < 1)$, and $\omega\prime$, $\psi\prime$ are the parameters of the Critic and Actor target networks, respectively, which are continuously adjusted by a soft update strategy:

$$\omega\prime \leftarrow \tau\omega + (1-\tau)\omega\prime, \psi\prime \leftarrow \tau\psi + (1-\tau)\psi\prime \tag{15}$$

where $\tau \ll 1$ is the target network update rate coefficient. parameter updates for the Actor network are then realized by estimating the gradient through the Critic network:

$$\nabla_\psi J(\psi) = \mathbb{E}_{\mathbf{s}_t \sim D}[\nabla_\mathbf{a} Q_\omega(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi_\psi(\mathbf{s}_t)} \nabla_\psi \pi_\psi(\mathbf{s}_t)] \tag{16}$$

During the training process, the data distribution is ensured to be stable through the empirical playback mechanism to improve the training efficiency and generalization ability. The specific training algorithm process is described as shown in the following pseudo-code.

Input: Maximum training episodes $E_{\max}$, replay buffer size D, discount factor γ, learning rates η_ψ, η_ω, target network update rate τ
Output: Trained compression policy network π_ψ
Initialize Actor network π_ψ and Critic network Q_ω with parameters ψ, ω;
Initialize target networks: $\psi' \leftarrow \psi$, $\omega' \leftarrow \omega$;
Initialize replay buffer D;
for *episode* $= 1$ **to** $E_{\max}$ **do**
  Initialize environment state $\mathbf{s}_1$;
  for $t = 1$ **to** T **do**
    Sample action $\mathbf{a}_t = \pi_\psi(\mathbf{s}_t) + \mathcal{N}_t$ (exploration noise);
    Execute action $\mathbf{a}_t$, receive reward r_t and next state $\mathbf{s}_{t+1}$;
    Store transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ into buffer D;
    Randomly sample minibatch from D;
    Update Critic parameters ω by minimizing loss $\mathcal{L}(\omega)$;
    Update Actor parameters ψ by maximizing objective $J(\psi)$;
    Soft update target networks:
      $\omega' \leftarrow \tau\omega + (1-\tau)\omega'$;
      $\psi' \leftarrow \tau\psi + (1-\tau)\psi'$;

Algorithm 1. Training of transformer-based dynamic semantic compression strategy based on DDPG

Receiving end

In highly dynamic and multi-tasking intelligent transportation scenarios, the receiver not only needs to effectively reconstruct and decode the low-dimensional semantic latent variables transmitted from the sender, but also needs to deeply fuse them with local multimodal traffic data (e.g., roadside sensing information, vehicle self-awareness data) to recover more complete and discriminative scene semantics. To this end, this study proposes a multimodal fusion framework based on the combination of Transformer's decoding mechanism and graph neural network, which combines the received semantic latent variable sequences with local data to perform preliminary feature enhancement through the cross-modal attention mechanism, and then, on the constructed spatial semantic graph, the potential interactions and contextual dependencies between different traffic entities are mined by using the multi-head graph attention mechanism to obtain a more refined global semantic representation. To cope with the coupling and competition between multiple concurrent tasks, this paper further proposes a multi-task optimization engine that combines reinforcement learning and self-supervised perturbation signals to achieve synergistic performance enhancement by dynamically adjusting the task weights. The receiver architecture takes semantic reconstruction and multimodal fusion as the core, taking into account the robustness and scalability in complex traffic environment. The processing flow of the receiving end is shown in Fig. 3.

Transformer-graph neural network multimodal fusion framework

In the aforementioned sender-side phase, we propose a dynamic semantic compression mechanism based on CVAE with Transformer-DRL, which enables multimodal data in intelligent transportation system scenarios to be efficiently and real-time encoded into compact, low-dimensional semantic latent variable representations, and to be adaptively compressed and transmitted to overcome the fluctuation of complex network environments. However, how to further decode and fuse these highly compressed semantic data efficiently at the receiver side for multi-task traffic scenario understanding, such as accident detection, behavior recognition, and scenario semantic understanding, remains an important challenge in current research. Therefore, we further propose a multimodal fusion framework based on Transformer and Graph Neural Network (Transformer-GNN) to efficiently decode and fuse the compressed semantic data transmitted at the sender's end to support comprehensive semantic understanding at the receiver's end.

The receiver will receive a sequence of semantic latent variables processed by the Transformer-DRL dynamic compression mechanism at the sender, denoted as $\mathbf{Z}^{comp} = [\mathbf{z}_1^{comp}, \mathbf{z}_2^{comp}, \ldots, \mathbf{z}_T^{comp}]$, where each $\mathbf{z}_t^{comp} \in \mathbb{R}^{d_z}$ is the compressed semantic representation of the corresponding moment. Although this compressed representation reduces the transmission, there is still the problem of insufficient information for direct use in traffic scene understanding due to its high compression. Therefore, the compressed semantics needs to be effectively decoded and semantically fused with local multimodal traffic environment data, such as local vehicle state data or roadside unit state data, at the receiver side to obtain a more complete scene understanding. We utilize the Transformer decoder structure to perform preliminary semantic decoding of the compressed semantic latent variable sequence $\mathbf{Z}^{comp}$ to recover a rich semantic contextual feature representation.The decoding of semantic information and context enhancement is achieved internally in the Transformer decoder

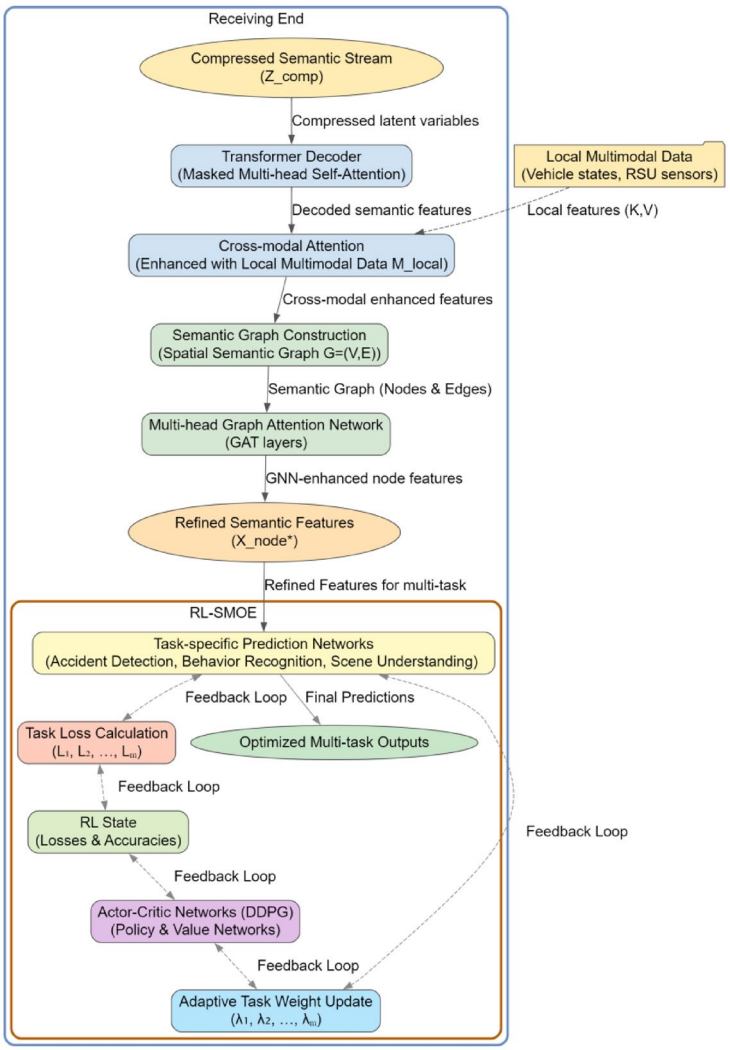

Fig. 3. Flowchart of receiving end processing.

through the masked multi-head self-attention and cross-modal attention mechanisms. The decoding process is represented as:

$$\mathbf{Z}^{dec} = \mathrm{TD}(\mathbf{Z}^{comp}, \mathbf{M}_{local}) \tag{17}$$

where TD denotes the Transformer Decoder and $\mathbf{M}_{local}$ denotes the set of modal feature information collected from the local traffic environment, which is used to assist the decoder to further enhance the semantic information. Cross-modal Attention is defined as:

$$\mathrm{CrossAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{18}$$

where $\mathbf{Q}$ comes from the Query of latent variable features within the decoder, and $\mathbf{K}, \mathbf{V}$ comes from the local modal data feature $\mathbf{M}_{local}$. By this step, the compressed semantic latent variable sequences are able to form a preliminary semantic feature fusion with the local modal data.

Then we further introduce the graph neural network (GNN) module to explicitly model the complex spatial and semantic relationships among traffic entities by constructing a spatial semantic graph. The spatial semantic graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ of the graph represents different modal data and traffic entities, such as vehicle nodes and roadside unit nodes, and the edge set E represents the spatial semantic association relationships between nodes, such as distance and interaction relationships. Each node feature is provided by the output feature representation $\mathbf{Z}^{dec}$ of the Transformer decoder, defined as:

$$\mathbf{X}_{node}^{(0)} = \mathbf{Z}^{dec}\mathbf{W}_{node} \tag{19}$$

where $\mathbf{W}_{node}$ is the trainable node feature mapping matrix. The initial adjacency matrix $\mathbf{A}$ between nodes is then determined based on the spatial constraints of the traffic scene:

$$\mathbf{A}_{ij} = \begin{cases} 1, Spatial distance d_{ij} between node i and node j < \epsilon \\ 0, \text{Other situations} \end{cases} \tag{20}$$

Because solely relying on the initial features of nodes and shallow topological relationships remains insufficient to fully uncover the complex and rich spatial semantic interactions among multimodal data nodes. Therefore we incorporate a multi-layer graph convolution network with graph attention mechanism to enhance the propagation of node features on the spatial semantic graph. For any node v_i, the feature propagation rule of the $l+1$ th layer graph convolution is defined as:

$$\mathbf{x}_i^{(l+1)} = \sigma\Big(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{x}_j^{(l)}\Big) \tag{21}$$

where $\mathbf{x}_i^{(l)} \in \mathbb{R}^{d_l}$ denotes the features of node v_i at layer , $\mathbf{W}^{(l)}$ is the trainable weight matrix, $\mathcal{N}(i)$ is the set of neighboring nodes of node v_i, $\sigma(\cdot)$ is the nonlinear activation function, and $\alpha_{ij}^{(l)}$ is the graph attention weight coefficient, which is computed in a way that introduces the attention mechanism in Transformer:

$$\alpha_{ij}^{(l)} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}^{(l)}\mathbf{x}_i^{(l)} \parallel \mathbf{W}^{(l)}\mathbf{x}_j^{(l)}]))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}^{(l)}\mathbf{x}_i^{(l)} \parallel \mathbf{W}^{(l)}\mathbf{x}_k^{(l)}]))} \tag{22}$$

where $\mathbf{a}$ is the parameter vector of the attention mechanism and $\parallel$ denotes the vector splicing operation. Through this Transformer-based graph attention mechanism, each node can adaptively and dynamically adjust the degree of fusion to the information of neighboring nodes according to the semantic correlation and topological relationship between nodes.

In order to further improve the stability and generalization of the feature propagation process, we introduce the structure of multi-head graph attention mechanism, i.e., the feature splicing and fusion of multiple independent graph attention heads, and the updating formula of each node's feature is defined as:

$$\mathbf{x}_i^{(l+1)} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{(l,k)} \mathbf{W}^{(l,k)} \mathbf{x}_j^{(l)}\Big) \tag{23}$$

where K is the number of multi-head attention heads, $\mathbf{W}^{(l,k)}$ and $\alpha_{ij}^{(l,k)}$ denote the weight matrix and attention coefficients of the k-th attention head, respectively.

After the above multi-layer graph convolutional feature propagation, each node feature on the spatial semantic graph will fuse all kinds of modal information and spatial semantic relations in the traffic scene, thus obtaining the final node feature representation $\mathbf{x}_i^*$.

Reinforcement learning self-supervised multi-task optimization engine

In practical applications of intelligent transportation systems, it is often necessary to simultaneously handle multiple complex semantic understanding tasks, such as traffic accident detection, driver behavior recognition, and traffic scene semantic understanding. The accident detection task mainly focuses on the real-time identification of traffic accidents, such as collisions, sudden braking, and other abnormal behaviors, while the behavior recognition task emphasizes the analysis of driver behavior patterns, such as changes in speed and lane switching. Although the goals of these two tasks are different, they share significant similarities in data processing and semantic understanding, and they are complementary. Accident detection can predict traffic accidents by identifying abnormal driver behaviors, while behavior recognition can optimize its accuracy by leveraging the background information from accident detection. Based on the principle of shared representation in multi-task learning, accident detection and behavior recognition tasks can be co-optimized within the same model, sharing underlying features and intermediate information to enhance overall performance. These tasks exhibit semantic correlations and potential conflicts. The key challenge in the field of intelligent transportation system research is how to achieve collaborative optimization of multiple tasks within a unified framework, avoid interference between tasks, and improve overall understanding performance.

In the method proposed in this paper, we introduce a Reinforcement Learning-based Self-supervised Multi-task Optimization Engine (RL-SMOE), which automatically adjusts the weight allocation of the multi-task network through a self-supervised mechanism to achieve collaborative optimization and adaptive adjustment of multi-task performance. By combining reinforcement learning with self-supervision, RL-SMOE can dynamically adjust task weights based on the characteristics of each task, fully leveraging the complementarity between tasks, maintaining balance during joint optimization, and avoiding interference between tasks. In this way, the system can improve the independent performance of each task while performing multi-task learning, achieving efficient collaborative optimization in complex traffic scenarios.

In this phase, we first formalize the definition for the multi-task learning framework, and set the multi-task set as $\mathcal{T} = \{T_1, T_2, \ldots, T_M\}$, which corresponds to the M semantic understanding tasks that are concerned by the intelligent transportation system, such as traffic accident detection, vehicle behavior recognition, and scene semantic understanding. The final feature representation of a node obtained by the Transformer-GNN fusion

framework is $\mathbf{X}^* = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*\}$, where N denotes the number of nodes on the spatial semantic graph. These node features are fed into each task-specific prediction network separately to obtain the corresponding predicted output $\hat{y}_i^{(m)}$ for each task. The loss function corresponding to each task is defined as:

$$\mathcal{L}_m(\theta_m) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{task}(y_i^{(m)}, \hat{y}_i^{(m)}), m \in \{1,2,\ldots,M\} \tag{24}$$

where θ_m denotes the model parameter $y_i^{(m)}$ corresponding to the m-th task denotes the true label, $\hat{y}_i^{(m)}$ is the corresponding predicted output, and $\mathcal{L}_{task}$ is the task-specific loss function. It is straightforward to train the joint loss function for multiple tasks, i.e.:

$$\mathcal{L}_{multi}(\theta) = \sum_{m=1}^{M}\lambda_m\mathcal{L}_m(\theta_m) \tag{25}$$

where λ_m is the weight coefficient of task m. In order to effectively deal with the dynamic interaction between tasks and the difference in optimization difficulty, we propose an adaptive adjustment mechanism for task weights based on the joint drive of reinforcement learning and self-supervision mechanism to dynamically optimize the weight coefficients λ_m. We define the state of the intelligent body as a state vector composed of the current training loss of each task and the model performance index:

$$\mathbf{s}_t = [\mathcal{L}_1^{(t)}, \mathcal{L}_2^{(t)}, \ldots, \mathcal{L}_M^{(t)}, ACC_1^{(t)}, ACC_2^{(t)}, \ldots, ACC_M^{(t)}] \tag{26}$$

where $\mathcal{L}_m^{(t)}$ and $ACC_m^{(t)}$ denote the loss and performance accuracy of the m-th task in the t-th round of training, respectively. The action $\mathbf{a}_t$ is then defined as the strategy vector for adjusting the task weight λ_m:

$$\mathbf{a}_t = [\Delta\lambda_1^{(t)}, \Delta\lambda_2^{(t)}, \ldots, \Delta\lambda_M^{(t)}] \tag{27}$$

The policy actions of each round of reinforcement learning will dynamically adjust the weight allocation of each task, and the specific task weight update rule is:

$$\lambda_m^{(t+1)} = \lambda_m^{(t)} + \Delta\lambda_m^{(t)}, \text{s.t.}\sum_{m=1}^{M}\lambda_m^{(t+1)} = 1, \lambda_m^{(t+1)} \geq 0 \tag{28}$$

The reward function, on the other hand, is determined by the degree of improvement in the joint multitasking performance metric, defined as:

$$r_t = \sum_{m=1}^{M}\beta_m\left(\frac{ACC_m^{(t+1)} - ACC_m^{(t)}}{ACC_m^{(t)}}\right) - \gamma\sum_{m=1}^{M}|\Delta\lambda_m^{(t)}| \tag{29}$$

where β_m is the reward coefficient of each task, and γ is the task weight adjustment penalty term, which is used to constrain the action magnitude. Next, we adopt Deep Deterministic Policy Gradient (DDPG) algorithm as the core training framework for multi-task weight control. DDPG adopts a joint architecture of Actor network and Critic network, where the Actor network (π_ϕ) takes the multi-task's current state vector $\mathbf{s}_t$ as an input , outputs the task weight adjustment action $\mathbf{a}_t$, while the Critic network (Q_ψ) is responsible for evaluating the long-term value of the Actor network's output action, and the specific evaluation function is expressed as:

$$Q_\psi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|\mathbf{s}_t, \mathbf{a}_t] \tag{30}$$

where the discount factor $\gamma \in (0,1)$ is used to balance the long-term and short-term rewards. The training objective of the Critic network is to minimize the Bellman error, with a specific loss function of:

$$\mathcal{L}_{Critic}(\psi) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\sim\mathcal{D}}[(r_t + \gamma Q_{\psi\prime}(\mathbf{s}_{t+1}, \pi_{\phi\prime}(\mathbf{s}_{t+1})) - Q_\psi(\mathbf{s}_t, \mathbf{a}_t))^2] \tag{31}$$

where $\mathcal{D}$ is the empirical playback buffer, and $\psi\prime$,$\phi\prime$ denote the target network parameters corresponding to the Critic and Actor networks, respectively, which are updated by a soft update strategy:

$$\psi\prime \leftarrow \tau\psi + (1-\tau)\psi\prime, \phi\prime \leftarrow \tau\phi + (1-\tau)\phi\prime \tag{32}$$

The parameters of the Actor network are then updated by maximizing the Q-value of the output of the evaluation network, i.e., by gradient ascent:

$$\nabla_\phi J(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}[\nabla_{\mathbf{a}}Q_\psi(\mathbf{s}_t, \mathbf{a})|_{\mathbf{a}=\pi_\phi(\mathbf{s}_t)}\nabla_\phi\pi_\phi(\mathbf{s}_t)] \tag{33}$$

To further improve the generalization ability and robustness of the policy network to the multi-task learning state space during training, we innovatively introduce a self-supervised signal to guide reinforcement learning policy optimization. The self-supervised mechanism generates pseudo-task samples based on the perturbation method in the task feature space, and calculates the loss value of the pseudo-task as an additional self-supervised reward signal r_{self}, defined as:

$$r_{self}^{(t)} = -\sum_{m=1}^{M} |\mathcal{L}_m^{pseudo,(t)} - \mathcal{L}_m^{(t)}| \tag{34}$$

where $\mathcal{L}_m^{pseudo,(t)}$ is the loss of the pseudo task under the perturbed feature condition and $\mathcal{L}_m^{(t)}$ is the loss of the real task. This self-supervised reward signal encourages the policy network to select those actions that are more stable under the task perturbations, thus improving the generalization performance of the policy network in complex environments. Thus the overall reward function of the reinforcement learning algorithm is finally expressed as:

$$R_{total}^{(t)} = r_t + \eta r_{self}^{(t)} \tag{35}$$

where η is the weight coefficient of the self-supervised reward signal.

Based on the above process, we propose the complete multi-task reinforcement learning self-supervised optimization algorithm flow with the following pseudo-code.

Input: Maximum training episodes $E_{\max}$, replay buffer size $\mathcal{D}$, discount factor γ, self-supervised reward weight η, target network update rate τ

Output: Trained multi-task adaptive optimization policy network π_ϕ

Initialize Actor network π_ϕ and Critic network Q_ψ with parameters ϕ, ψ;
Initialize target networks: $\pi_{\phi'} \leftarrow \pi_\phi$, $Q_{\psi'} \leftarrow Q_\psi$;
Initialize replay buffer $\mathcal{D}$;
for *episode* $= 1$ **to** $E_{\max}$ **do**
  Observe multi-task current state $\mathbf{s}_1$;
  for $t = 1$ **to** T **do**
    Select action $\mathbf{a}_t = \pi_\phi(\mathbf{s}_t) + \epsilon_t$ using Actor network and exploration noise ϵ_t;
    Execute action $\mathbf{a}_t$, update task weight $\lambda_m^{(t+1)}$, observe new state $\mathbf{s}_{t+1}$;
    Compute reward r_t and self-supervised reward $r_{\text{self}}^{(t)}$;
    Store transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, r_{\text{self}}^{(t)}, \mathbf{s}_{t+1})$ into buffer $\mathcal{D}$;
    Randomly sample minibatch from buffer $\mathcal{D}$;
    Update Critic parameters ψ by minimizing loss $\mathcal{L}_{\text{Critic}}(\psi)$;
    Update Actor parameters ϕ by maximizing objective $J(\phi)$;
    Soft update target networks:
      $\psi' \leftarrow \tau\psi + (1-\tau)\psi'$;
      $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$;

Algorithm 2. Reinforcement learning-based self-supervised multi-task optimization (RL-SMOE)

Experimental analysis

Experimental setup

In order to objectively and effectively verify the performance of the SMART mechanism proposed in this study, we built a perfect experimental hardware and software environment and carried out detailed experimental parameter settings. In terms of hardware platform, the experimental environment is built on a high-performance server equipped with an Intel Xeon Silver 4214 processor (2.20 GHz), 128 GB of RAM, and two NVIDIA RTX 3090 GPU graphics cards, with a single GPU card with 24 GB of graphics memory to meet the high demand for computing power during the multimodal data processing and deep learning model training. Resources during multimodal data processing and deep learning model training. To improve the stability of the training process, we configure an additional 4 TB solid state disk for storing raw data and a large amount of intermediate feature data generated during the experiment.

As for the software environment, the experimental system is based on Ubuntu 20.04 LTS operating system, the algorithm development and model training are mainly realized based on Python 3.8, the deep learning framework is based on PyTorch 1.12.0 and CUDA 11.3, and PyTorch Geometric 2.1.0 is used as the framework for the development of graphical neural network and combined with the Transformers 4.21.1 library to realize the construction of the Transformer model, and the reinforcement learning part is realized based on the Stable-Baselines3 1.5.0 framework. During the training process of the SMART method, we initially set the model learning rate to 1e-4 and use the cosine annealing scheduling strategy for adaptive dynamic adjustment. The latent variable dimension of the CVAE model was set to 256, the number of Transformer module attention heads was set to 8, and the feature dimension of each head was 64. The discount factor γ was set to 0.99 during the reinforcement learning training process, the capacity of the experience playback buffer pool was set to 100,000 experiences, the batch size was set to 256, and the target network update rate τ was set to 0.005. The temperature parameter of the comparative learning loss function in the SMART method is taken as 0.07, and the self-supervised reward weighting factor η is set to 0.5

In this experiment, we use several key clips with complete multimodal information (camera images, radar, LIDAR point cloud, and odometer information) from nuScenes to focus on urban roadway and intersection scenarios with relatively dense traffic flow and rich events. In order to adapt the semantic collaborative transmission mechanism to the video data, we uniformly downscale the camera frame rate to 10 frames per second and synchronize the radar and LIDAR data sampling rate to 10 Hz. After aligning with the camera frames, we organize the image frames, radar echoes, and LIDAR point cloud data at the same moment into multimodal samples. The total number of selected multimodal samples is about 50,000, and they are split into training, validation, and test sets in the ratio of 70%:15%:15%.

To validate the performance of the SMART method, we set up an experimental comparison group to compare the performance using two representative comparison algorithms in the current semantic communication and Telematics field, namely DeepSC[23] and SSS[21] methods. In the performance evaluation of this study, we compare the actual effectiveness of the three semantic communication methods, SMART, DeepSC, and SSS, in terms of four key metrics: semantic similarity, transmission efficiency, robustness, and latency. Semantic Fidelity focuses on measuring how well the original semantic information X fits the reconstructed information $\hat{X}$ in terms of meaning, and a metric function based on vector similarity can be defined:

$$\mathrm{Sim}(X, \hat{X}) = \frac{\mathbf{h}(X) \cdot \mathbf{h}(\hat{X})}{\| \mathbf{h}(X) \| \| \mathbf{h}(\hat{X}) \|} \tag{36}$$

where $\mathbf{h}(\cdot)$ denotes the high-dimensional semantic embedding vector extracted using the language model or deep network.

Transmission Efficiency defines the relationship between the amount of semantic information SI and the bandwidth B, and the duration Δt, denoted as:

$$\mathrm{SSE} = \frac{\mathrm{SI}}{B \cdot \Delta t} \tag{37}$$

It is used to measure the amount of semantic information successfully delivered per unit of bandwidth and time.

Robustness is mainly reflected in the stability of semantic transmission under different channel states, noise levels or packet loss rates, and is quantified using the Semantic Success Rate (SSR) metric, which is defined as the percentage of times that a certain threshold similarity is reached in N transmissions:

$$\mathrm{SSR} = \frac{1}{N} \sum_{i=1}^{N} 1[\mathrm{Sim}(X_i, \hat{X}_i) \geq \tau] \tag{38}$$

where τ is the threshold for semantic similarity determination.

Latency Performance consists of the sum of encoding delay, channel transmission delay and decoding delay:

$$\mathrm{Latency}_{\mathrm{total}} = T_{\mathrm{encode}} + T_{\mathrm{channel}} + T_{\mathrm{decode}} \tag{39}$$

Latency can be particularly concerned with whether the total end-to-end elapsed time can meet the real-time requirements of the corresponding scenarios of the intelligent transportation system (e.g., emergency alerts, autonomous driving decisions).

Algorithm complexity analysis

To illustrate the deployability of SMART on real-world in-vehicle/roadside hardware, we evaluate the computational complexity and inference latency of the four core sub-modules of the system—CVAE-Encoder + Transformer-DRL compression, Transformer Decoder, GNN multimodal fusion, and RL-SMOE task scheduling. Transformer Decoder, GNN Multimodal Fusion, and RL-SMOE Task Scheduling—were evaluated in terms of computational complexity and inference latency. The tests were conducted on a 30 W-class Jetson AGX Xavier (8 GB LPDDR4X) and a desktop-class RTX 3090 (24 GB GDDR6X), with inputs of a single-frame RGB image of 1280 × 720, a radar bird's-eye view of 256 × 256, and a LIDAR point cloud of 15,000 points. All modules first count the number of parameters and theoretical FLOPs at FP32, and then measure the end-to-end delay at RTX 3090 with TensorCore FP16 and Xavier with TensorRT INT8 inference. The experimental results are shown in Table 1.

Module	Number of participants (M)	Theoretical FLOPs (G)	RTX 3090 Inference latency (ms)	Jetson AGX Xavier Inference latency (ms)
CVAE-Encoder + T-DRL Compression	11.8	13.2	7.1	20.5
Transformer Decoder	8.1	9.0	5.2	13.6
GNN Multimodal Fusion	3.9	5.0	3.0	8.4
RL-SMOE Task Scheduling Header	2.0	1.8	1.9	4.8
Total	25.8	29.0	17.2	47.3

Table 1. Computational complexity and inference latency of each sub-module of SMART.

Model name	Number of participants (M)	Theoretical FLOPs (G)	RTX 3090 inference latency (ms)	Jetson AGX Xavier inference latency (ms)
SMART	25.8	29.0	17.2	47.3
DeepSC	23.4	25.0	20.5	55.0
SSS	18.2	20.0	22.3	60.0

Table 2. Comparison of the number of parameters and computational complexity of SMART with DeepSC and SSS.

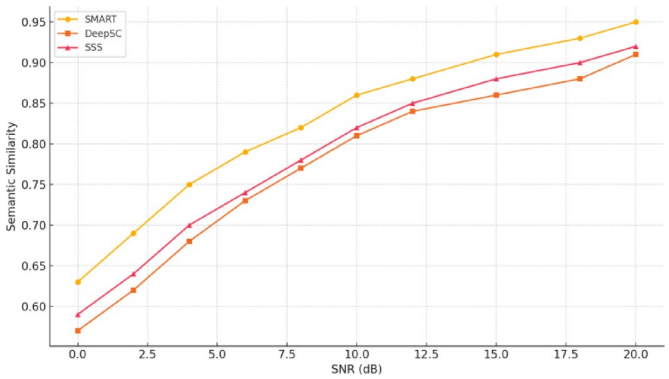

Fig. 4. Comparison of Semantic Similarity of the three methods under different SNR conditions.

In the desktop environment with an RTX 3090, the SMART system takes an average of 17 ms from receiving the multimodal packets to generating fused features and multi-task predictions, corresponding to a perception frequency of 58 Hz, which is above the commonly used industry safety threshold of 25 Hz. When ported to a 30 W automotive-grade SoC (Jetson AGX Xavier), the end-to-end latency remains around 47 ms, equivalent to a frame rate of 21 Hz, meeting the real-time requirements of a 50 ms closed-loop at speeds below 50 km/h in urban conditions. The system's memory usage is 2.0 GB (FP16), still under the 25% threshold of the 8 GB LPDDR4X memory of Xavier, indicating the algorithm's potential for direct deployment on the vehicle. Further analysis of the submodule contributions reveals that the CVAE-Encoder and Transformer Decoder consume about 75% of the total computational power, being the main sources of latency and power consumption. The GNN fusion and RL-SMOE account for the remaining quarter, with limited impact on overall delay. This suggests that to run on lower-power platforms, structural pruning or operator replacement should first be applied to the encoding and decoding stages. After applying 40% channel pruning and enabling full INT8 quantization, the model size reduces to 18.3 M parameters and 17.4 G FLOPs, with inference latency on Xavier dropping to 32 ms. The average decrease in core metrics such as semantic similarity and SSR is only 1.2%, which is still within the 1.5% engineering tolerance range, proving the model's robustness to lightweight operations.

We also performed a complexity comparison between the SMART method and two representative semantic communication models, with experimental data shown in Table 2. In terms of parameter count and FLOPs, SMART has 25.8 million parameters and 29.0G FLOPs, slightly higher than DeepSC (23.4 million parameters, 25.0G FLOPs) and SSS (18.2 million parameters, 20.0G FLOPs). This indicates that SMART requires more computational resources when handling multimodal data, but its higher parameter count and FLOPs also reflect its capability to process complex semantic information, particularly in high-dimensional data compression and transmission.

In terms of inference latency, SMART achieves 17.2 ms on the RTX 3090 platform, significantly lower than DeepSC's 20.5 ms and SSS's 22.3 ms, demonstrating that SMART performs excellently on high-performance hardware, providing faster real-time responses. On the Jetson AGX Xavier embedded platform, SMART's inference latency is 47.3 ms, slightly higher than on the RTX 3090, but still superior to DeepSC (55.0 ms) and

SSS (60.0 ms), indicating SMART's good adaptability to in-vehicle platforms and its ability to meet the real-time demands of automotive systems.

Experimental results and analysis

The experimental results of this study comparing the two algorithms with DeepSC and SSS in terms of semantic similarity metrics are shown in Fig. 4. In order to be close to the diversity of real wireless communication environments and to simulate different noise levels, this experiment is tested at multiple take-off points with SNR from 0 to 20 dB, and the average semantic similarity of the three methods is counted at each take-off point. Figure 4 shows the average semantic similarity results of the three semantic communication methods at different SNRs. Overall, the semantic similarity of the three methods shows an increasing trend with the increasing SNR value, which indicates that the system is easier to accurately restore the semantics of multimodal data in traffic scenarios under better channel conditions.

When the channel environment is harsh (0–4 dB), the average semantic similarity of the three methods generally ranges from 0.57 to 0.75. In contrast, SMART reaches 0.63 at 0 dB, while DeepSC and SSS are only 0.57 and 0.59, respectively. In contrast, SMART reaches 0.63 at 0 dB, while DeepSC and SSS are only 0.57 and 0.59, respectively, and at 4 dB, SMART has improved to 0.75, which is about 5–7 percentage points ahead of the two methods. The main reason is that the transmitter in SMART compresses the multimodal data with a self-supervised conditional variational self-encoder, which preserves the key semantic information. The dynamic regulation mechanism based on Transformer and DRL can adjust the transmission strategy in time to reduce semantic loss under severe noise environment. DeepSC, on the other hand, has better semantic fidelity at the sentence level, but is relatively weak in multimodal fusion. SSS is more inclined to spectral optimization, which makes it difficult to take care of deep semantic extraction in extreme noise.

In the medium SNR range (6–12 dB), the semantic similarity of the three methods increases significantly: DeepSC increases from 0.73 at 6 dB to 0.84 at 12 dB, SSS increases from 0.74 to 0.85, and SMART increases from 0.79 to 0.88, maintaining an overall lead of about 4–5 percentage points. At this time, DeepSC and SSS make up for some of their previous weaknesses in multimodal semantic fusion or spectrum resource scheduling under better communication conditions, but the receiver side of SMART, relying on the Transformer-GNN framework, is obviously more capable of fusing traffic data from multiple sources, and with the support of self-supervised multitasking optimization in multi-tasking contexts, such as accident detection and vehicle behavior recognition, it obtains more refined semantic restoration effects. finer semantic reduction effects.

In the higher SNR region (15–20 dB), when the SNR exceeds 15 dB, the semantic similarity of the three methods is already higher than 0.85, among which SMART reaches 0.91 at 15 dB, and 0.95 at 20 dB, while DeepSC and SSS stay at the level of 0.86–0.91 and 0.88–0.92, respectively. Under high SNR conditions, the noise impact is relatively limited, DeepSC can better utilize its textual semantic coding and decoding advantages, and SSS has more usable experience in spectrum sharing and transmit power control level in Telematics. However, SMART still maintains a 3–4 percentage point lead through adaptive compression and accurate modeling of multimodal semantic information by graph attention network, which proves that the system also has outstanding performance in semantic extraction, dynamic transmission, and multimodal fusion under high channel quality environment.

In order to more comprehensively examine the actual performance of the system under different network load levels, we constructed two scenarios of medium-load and high-load environments, and tested the SSE values of SMART, DeepSC, and SSS under different SNRs. The data samples and the multimodal preprocessing process of each experiment are consistent with the previous one, but focus on simulating the transmission demand under relatively abundant bandwidth and bandwidth-constrained, high-concurrency scenarios, respectively. The experimental results are shown in Fig. 5.

For the medium load scenario, it is assumed that the bandwidth B is moderate (10 MHz), the traffic data is relatively not extremely congested, and the system needs to serve several vehicle-side and road-side units at the same time, but the overall traffic volume is still within the controllable range. Five typical channel levels with SNR = 6, 10, 14, 18, and 20 dB are selected for testing, and the results are shown in Fig. 5.

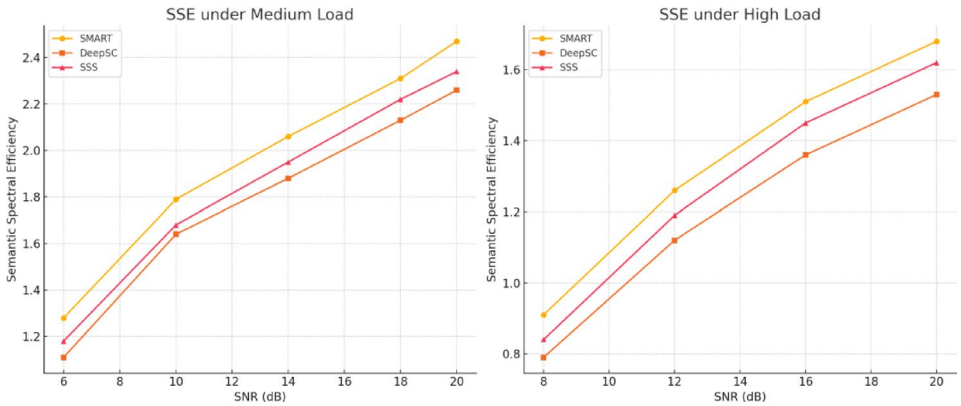

Fig. 5. Comparison of SSE of three algorithms.

At SNR = 6 dB, all three approaches are limited by the noise level, which makes the semantic transmission efficiency of the available bandwidth low. However, SMART balances the fidelity of multimodal key semantic extraction with compression rate thanks to its transmitter-side self-supervised CVAE compression strategy and reinforcement-learning-driven dynamic bandwidth allocation, and reaches an SSE of 1.28, which is higher than that of DeepSC (1.11) and SSS (1.18). As the channel environment improves (SNR gradually increases from 10 to 20 dB), the SSEs of all three methods increase significantly, with DeepSC and SSS reaching the upper performance limit around 2.26 and SMART reaching 2.47. This indicates that SMART can make full use of the available bandwidth when the channel quality improves to compress the transmission with multimodal Transformer-DRL compression in a medium load scenario. This indicates that SMART can make full use of the available bandwidth when the channel quality is improved, and further eliminate data redundancy and improve the effective transmission of semantic information by using multimodal Transformer-DRL compressed transmission and receiver-side graph neural network fusion techniques.

For the high load scenario, it is assumed that the network bandwidth B is tighter (e.g., 5 MHz), while the traffic flow and the number of terminals are significantly increased to simulate large-scale concurrent data requests in situations such as highly congested roads or unexpected events. Due to the more extreme environment, we chose four levels of relatively moderate to good (8 dB, 12 dB, 16 dB, 20 dB) channel conditions to test SSE and the results are shown in Fig. 5.

In the context of significant bandwidth constraints, the overall SSE values of the three methods are significantly lower than in the medium load case, but SMART still maintains the highest transmission efficiency at all SNR points. At 8 dB, SMART is 0.91, higher than DeepSC (0.79) and SSS (0.84). At SNR = 20 dB, the difference between SMART and the two remains above 0.06. Although DeepSC's text-oriented semantic compression can improve the transmission efficiency to a certain extent when the bandwidth is tight, the semantic expression of multimodal information is not as detailed as that of SMART, and although SSS has excellent results in spectrum sharing and power control, it does not do more refined self-supervised/comparative learning for multimodal semantic compression, so SSE is still slightly inferior to SMART.

In order to reflect the impact of different communication environment interference levels on robustness, we mainly examine the variation of packet loss rate (PLR) in the range of 0% to 20%, and in the process, we count the average SSR of the three semantic communication methods (SMART, DeepSC, and SSS) on the test set. The packet loss rate is simulated by randomly discarding part of the packets in the transmission channel, and the ratio of lost packets to the total number of transmitted packets is the PLR. The ratio of lost packets to the total number of packets sent is the PLR. Figure 6 gives the experimental results under different packet loss rate conditions, where other network conditions such as bandwidth, SNR, and traffic levels are kept in the medium range (SNR = 10 dB, network bandwidth B = 10 MHz) to reduce the cross-factor interference.

PLR = 0% ~ 5% low packet loss rate interval, under the condition of good network environment, the SSR of the three methods are kept between 0.80 and 0.92, which can complete the semantic transmission stably. At this time, SMART reaches 0.92 and 0.88, which is about 6–8 percentage points higher than DeepSC and 4–6 percentage points higher than SSS, respectively. Since the system delivers most multimodal packets successfully due to the small chance of packet loss, SMART has a slight advantage in SSR due to better error correction and redundancy utilisation of the few lost packets through self-supervised CVAE compression and enhanced multimodal fusion at the receiver side. The impact of network jitter and data loss increases significantly for packet loss rates between 10 and 20%. DeepSC drops from 0.73 to 0.60, SSS from 0.76 to 0.65, while SMART is relatively flat, slipping from 0.83 to 0.71, with the lead remaining at 5–11 percentage points. The reason is that SMART adopts Transformer-DRL, which can dynamically adjust the key semantic encoding rate of the transmitter to prioritise the transmission of core elements when packet loss increases. The receiver-side graph neural network is able to compensate the local voids caused by lost packets to some extent based on multimodal multi-source information. PLR ≥ 25% High packet loss rate interval, under the most extreme network environment, the SSR of the three methods comprehensively declines, but SMART can still be maintained in the range of 0.56–0.63, which is higher than DeepSC (0.44–0.52) and SSS (0.51–0.58). When the packet loss rate exceeds 25%, DeepSC focuses on semantic coding and decoding of textual content, which makes it difficult to recover the lost packets,

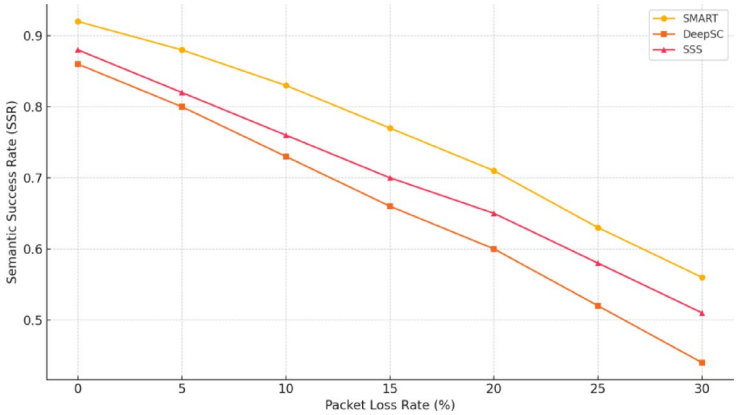

Fig. 6. Comparison of SSR of the three algorithms at different PLRs.

while SSS can dynamically adjust the spectrum and power, but it lacks a strategy for selecting key packets for multimodal data. In contrast, SMART's self-supervised and multi-tasked attention mechanism ensures that the core traffic scenario semantics are preserved with a small number of key coded packets in the event of a large number of packet losses.

In our experiments, we measure the performance of SMART, DeepSC and SSS in terms of end-to-end latency. We split the total delay into encoding delay (T_encode), channel transmission delay (T_channel) and decoding delay (T_decode), and finally count the total delay Latency_total. For comparison, the following four scenarios are based on the same experimental data size and multimodal setup, but differ in the number of concurrencies (load), SNR, and level so as to reflect the delay characteristics of the method under multiple combinations of low/high concurrencies and excellent/poor channels. The experimental results are shown in Fig. 7.

Low concurrency, moderate SNR (Scenario A), assuming bandwidth B = 10 MHz, SNR = 10 dB, relatively low vehicle concurrency (20 vehicles), and moderate network load and noise. Scenario A is selected from the nuScenes Boston-Seaport region daytime intersection segment, which has dense traffic and frequent occlusions, and is used as an urban intersection baseline scenario to evaluate the semantic collaboration performance in a typical signalized environment. We measure the average encoding delay (ms) at the sender of multimodal data, the average transmission delay (ms) under the current channel conditions, and the average decoding delay (ms) at the receiver of the three methods to obtain the total delay. In terms of coding delay, DeepSC takes slightly less time (18.7 ms) due to its focus on textual semantics and relatively straightforward coding process, while SMART takes more time (25.3 ms) due to its multimodal self-supervised CVAE and Transformer compression strategies. In terms of transmission delay, SMART's adaptive compression is effective in the current medium bandwidth and medium channel quality, reducing the amount of data and resulting in a lower T_channel (33.8 ms). DeepSC and SSS require greater transmission resources. Regarding decoding delay, SMART adopts a Transformer-GNN-based multimodal fusion and multi-task engine, involving deeper computations during decoding and resulting in 20.5 ms of processing. In contrast, DeepSC and SSS use relatively simpler decoding methods without multimodal fusion, but their processing time is more fragmented and ends up slightly higher in total. As for overall latency, the three methods are similar in this scenario (79.6–81.5 ms). Through joint optimization at both the sender and receiver sides, SMART significantly reduces the transmission time even though its encoding and decoding delay is somewhat higher, ultimately achieving slightly better total latency (79.6 ms).

Low concurrency, high SNR (Scenario B), compared with Scenario A, the channel is improved to SNR = 18 dB, and the rest of the conditions (concurrency, bandwidth) are unchanged, selected from the suburban two-lane section of the nuScenes Singapore-Holland-Village area, with a wide field of view and sparse road markings, to test SMART's generalization ability in low-obscurity, high-speed cruising scenarios. The main purpose is to test the generalization ability of SMART in low occlusion and high speed cruise scenarios. The delay performance of the three methods is evaluated in a high quality channel. As shown in the figure, the improvement in channel quality speeds up transmission, with the three methods' T_channel decreasing by 3–5 ms compared

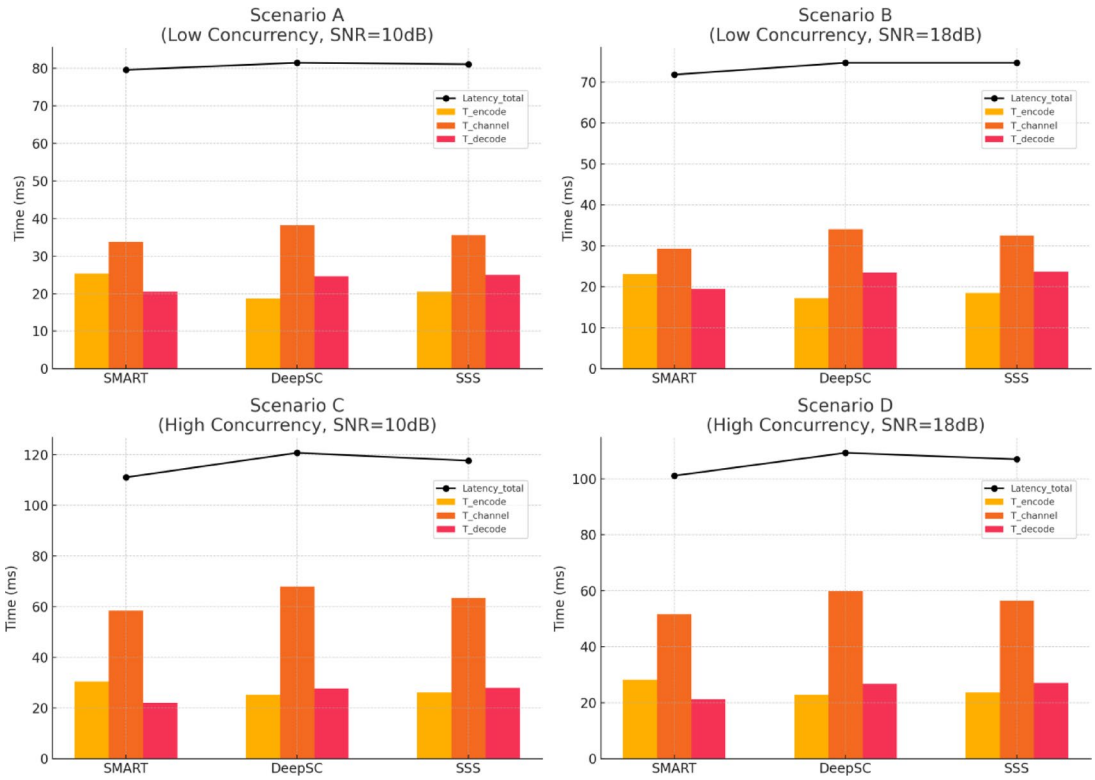

Fig. 7. Comparison of time delay.

Scenarios	Methods	Accident detection F1	Behavior recognition Top-1 Acc
Scenario A	DeepSC	0.821	0.892
	SSS	0.836	0.904
	SMART	0.881	0.931
	SeeUnsafe	0.855	0.915
Scenario B	DeepSC	0.803	0.876
	SSS	0.817	0.889
	SMART	0.862	0.918
	SeeUnsafe	0.845	0.910
Scenario C	DeepSC	0.785	0.861
	SSS	0.799	0.874
	SMART	0.847	0.906
	SeeUnsafe	0.830	0.895
Scenario D	DeepSC	0.731	0.823
	SSS	0.748	0.835
	SMART	0.801	0.872
	SeeUnsafe	0.785	0.860

Table 3. Comparison of accident detection and behavior recognition performance.

with Scenario A. SMART's adaptive compression once again brings advantages in both bandwidth utilization and transmission speed, reducing T_channel to 29.3 ms. However, because of the multimodal encoding and decoding process, the combined T_encode + T_decode remains slightly higher than the simpler single-modal or text-based methods. In terms of total latency, SMART clocks in at 71.8 ms, about 3 ms ahead of DeepSC and SSS (74.7 ms). For emergency scenarios or tasks with tight time requirements, this 3–5 ms difference can help improve response speed to a certain extent.

High concurrency, moderate SNR (Scenario C), compared with Scenario A, the number of concurrent vehicles is greatly increased (e.g., 80 vehicles), and the latency of all three is measured at SNR = 10 dB and bandwidth B = 10 MHz. This scenario is taken from the nuScenes Singapore-Onenorth Loop highway evening rush hour segment, with speeds of 80–110 km/h and frequent lane changes, and is used to test the stability of multimodal dynamic compression/scheduling under high concurrency and moderate channel conditions. As can be seen from the figure, the surge in concurrency leads to a rise in the total transmission demand, the channel competition is intense, and the T_channel of the three algorithms increases significantly (around 20–30 ms) compared to the low concurrency. In high-concurrency scenarios, the advantages of SMART's multimodal adaptive compression become even more pronounced. Although the encoding delay rises to 30.5 ms as a result, the effectively reduced transmission volume yields a lower T_channel compared with DeepSC and SSS, leading to a total latency of 111.0 ms—about 6–10% faster than the latter two methods. While DeepSC and SSS include certain optimizations for power or text processing, they lack a fine-grained multimodal compression and scheduling mechanism when dealing with a large number of concurrent terminals. As a result, they suffer more pronounced transmission bottlenecks, pushing total latency beyond 115 ms.

High concurrency and SNR (Scenario D), where concurrency remains high (80 vehicles) and SNR increases to 18 dB compared to Scenario A. This scenario was selected from the nuScenes Singapore-Queenstown nighttime moderate rainfall on the main roadway with significant degradation in visual sensing to evaluate the robustness of SMART under the challenges of both inclement weather and high concurrency. This is to evaluate the robustness of SMART under the dual challenges of bad weather and high concurrency. From the results in the figure, it can be seen that as the SNR increases, the transmission efficiency of the three methods increases significantly, but because the concurrency is still high, the T_channel is higher compared to the low concurrency. SMART is able to further reduce the occupied bandwidth in the context of multimodal CVAE compression and DRL optimisation at the transmitter side, with a T_channel of only 51.7 ms. Even though the combined T_encode + T_decode is about 2–4 ms higher than the other two methods, SMART still outperforms DeepSC (109.3 ms) and SSS (107.0 ms) with an overall total delay of 101.1 ms. SSS (107.0 ms).

From an overall perspective, SMART requires slightly more time in the encoding and decoding phases due to the extraction and fusion of multimodal features. However, in complex network environments, its adaptive compression and efficient transmission capabilities significantly reduce channel delay, resulting in a clear advantage in end-to-end latency.

To further quantify the effectiveness of SMART in the downstream safety tasks of accident detection and vehicle behavior recognition, we established a new set of multi-task evaluations on top of the existing four scenarios. The experiment uses the same batch of multimodal transmission outputs, which are decoded by the Transformer-GNN at the receiver end and directly fed into two lightweight prediction heads: Accident detection uses a binary classification ResNet-18, outputting accident/non-accident probabilities within a 1-s time window. Behavior recognition uses a temporal TCN + MLP, outputting six types of typical vehicle behaviors (straight, left/right turn, lane change, braking, acceleration). The training and testing follow a 70%/15%/15% split, with evaluation metrics using F1 score (for accident detection) and Top-1 accuracy (for behavior recognition). To compare the advantages of SMART in these two tasks, we selected three existing methods for comparison: DeepSC, SSS, and

SeeUnsafe. DeepSC and SSS are current mainstream methods in the field of semantic communication, while SeeUnsafe[37] is a solution specifically designed for traffic safety tasks, combining the advantages of computer vision and deep learning. The experimental results, shown in Table 3, present a performance comparison of each method across four different scenarios.

As shown in Table 3, SMART outperforms both DeepSC and SSS methods in accident detection and behavior recognition tasks, exhibiting higher F1 scores and Top-1 accuracy. In particular, SMART's performance improvement is especially significant in Scenarios A and B, fully demonstrating its advantages in multi-task semantic communication. Compared to SMART, SeeUnsafe performs slightly worse in some scenarios, which may be attributed to differences in its model architecture and task focus. SeeUnsafe primarily focuses on video classification and visual localization tasks, which limits its ability in multi-task collaborative optimization. Overall, the results confirm that SMART not only excels in link metrics but also maintains stable and significant performance advantages in safety–critical multi-task inference.

Conclusion

In this study, we focus on the problem of efficient transmission and semantic understanding of multimodal data in complex network environments for intelligent transport systems, and propose a multimodal semantic collaborative transmission mechanism that integrates the ideas of self-supervised learning and reinforcement learning. At the sending end, we use a self-supervised conditional variational self-encoder to represent and compress heterogeneous data from video, radar, and LIDAR at a deep semantic level, and use Transformer and deep reinforcement learning algorithms to achieve adaptive control of semantic dimensions in the transmission process, which significantly reduces bandwidth consumption and information loss caused by noise interference. At the receiving end, the Transformer decoder and graph neural network structure are fused together, using the correlation of multimodal spatial and temporal features to prevent data loss or noise interference, and supported by the reinforcement learning self-supervised multitasking optimization engine to achieve collaborative optimization of traffic accident detection, vehicle behaviour recognition, scene understanding and other multitasks.

Comprehensive experimental results in multiple dimensions show that, in terms of semantic similarity, SMART achieves higher semantic restoration than DeepSC and SSS under different channel conditions (SNR), especially in low and medium SNR environments, which proves that self-supervised semantic extraction and multimodal fusion strategies can effectively improve the system's noise immunity. In terms of transmission efficiency, SMART adopts the self-supervised CVAE fusion Transformer-DRL dynamic compression mechanism, which is able to deliver more and more accurate key semantic information under the same bandwidth and time limit. Even in medium- to high-concurrency scenarios, SMART maintains a comparatively significant lead. In terms of robustness, thanks to its multimodal redundancy design and reinforcement learning–based adaptive mechanism, SMART still achieves the highest semantic success rate (SSR) even when the packet loss rate (PLR) is increased to 20%–30%, substantially outperforming other methods. This demonstrates its adaptability to complex network fluctuations and random packet drops. Regarding latency, although SMART incurs a slight overhead in the multimodal encoding and decoding phases, its adaptive compression greatly reduces channel transmission time, resulting in lower end-to-end latency compared with DeepSC and SSS, particularly under high concurrency conditions where it exhibits a notable advantage.

In this paper, there are still a number of areas that deserve further exploration: on the one hand, this study mainly focuses on typical scenarios such as urban roads and intersections, and has not yet been validated on a large scale in larger-scale, geographically dispersed and more dynamically changing traffic networks. On the other hand, there is still much room for expanding the privacy protection and semantic security of multimodal data, as well as how to synergistically integrate with the ultra-low latency and edge computing (MEC) mechanisms of the 5G/6G ecosystem. In the future, we can consider combining the cutting-edge concepts of federated learning, differential privacy and multi-intelligence body reinforcement learning with SMART under the system of multi-vehicle collaboration and intelligent roadside perception fusion, so as to enhance the security, reliability and real-time adaptive capability of intelligent transport systems from a more macroscopic and multi-dimensional perspective.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author Shaojiang Liu on reasonable request via e-mail mrliuxinhua@xhsysu.edu.cn.

Received: 27 April 2025; Accepted: 6 August 2025
Published online: 14 August 2025

References

1. Zimmer, W., Wardana, G. A., Sritharan, S., et al. Tumtraf v2x cooperative perception dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 22668–22677 (2024).
2. Tan, J. et al. Beam alignment in mmWave V2X communications: A survey. *IEEE Commun. Surv. Tutor.* **3**(26), 1676–1709 (2024).
3. Bazzi, A. et al. On the design of sidelink for cellular V2X: A literature review and outlook for future. *IEEE Access* **9**, 97953–97980 (2021).
4. Salehi, B. et al. Deep learning on multimodal sensor data at the wireless edge for vehicular network. *IEEE Trans. Veh. Technol.* **71**(7), 7639–7655 (2022).
5. Azadani, M. N. & Boukerche, A. A novel multimodal vehicle path prediction method based on temporal convolutional networks. *IEEE Trans. Intell. Transp. Syst.* **23**(12), 25384–25395 (2022).
6. Geng, M. et al. Multimodal vehicular trajectory prediction with inverse reinforcement learning and risk aversion at urban unsignalized intersections. *IEEE Trans. Intell. Transp. Syst.* **24**(11), 12227–12240 (2023).

7. Wu, K. et al. Graph-based interaction-aware multimodal 2d vehicle trajectory prediction using diffusion graph convolutional networks. *IEEE Trans. Intell. Veh.* **9**(2), 3630–3643 (2023).
8. Feng, S. et al. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. Commun.* **12**(1), 748 (2021).
9. Chen, L. et al. Milestones in autonomous driving and intelligent vehicles: Survey of surveys. *IEEE Trans. Intell. Veh.* **8**(2), 1046–1056 (2022).
10. Cao, D. et al. Future directions of intelligent vehicles: Potentials, possibilities, and perspectives. *IEEE Trans. Intell. Veh.* **7**(1), 7–10 (2022).
11. Luo, X., Chen, H. H. & Guo, Q. Semantic communications: Overview, open issues, and future research directions. *IEEE Wirel. Commun.* **29**(1), 210–219 (2022).
12. Guo, S., Wang, Y., Zhang, N., et al. A survey on semantic communication networks: Architecture, security, and privacy. *IEEE Commun. Surv. Tutor.* (2024).
13. Liang, C. et al. Generative AI-driven semantic communication networks: Architecture, technologies and applications. *IEEE Trans. Cognit. Commun. Netw.* **1**(11), 27–47 (2024).
14. Ye, S., Wu, Q., Fan, P., et al. A survey on semantic communications in internet of vehicles. arXiv preprint arXiv:2503.03767, (2025).
15. Singh, M., Dubey, R. K. & Kumar, S. Vehicle telematics: An internet of things and big data approach. In *Artificial Intelligence and Machine Learning for EDGE Computing* 235–254 (Academic Press, 2022).
16. Xu, W. et al. Semantic communication for the internet of vehicles: A multiuser cooperative approach. *IEEE Veh. Technol. Mag.* **18**(1), 100–109 (2023).
17. Creß, C., Bing, Z. & Knoll, A. C. Intelligent transportation systems using roadside infrastructure: A literature survey. *IEEE Trans. Intell. Transp. Syst.* **25**(7), 6309–6327 (2023).
18. Khalil, R. A. et al. Advanced learning technologies for intelligent transportation systems: Prospects and challenges. *IEEE Open J. Veh. Technol.* **5**, 397–427 (2024).
19. Peelam, M. S. et al. A review on emergency vehicle management for intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **11**(25), 15229–15246 (2024).
20. Shao, Z., Wu, Q., Fan, P., et al. Semantic-Aware Resource Management for C-V2X Platooning via Multi-Agent Reinforcement Learning. arXiv preprint arXiv:2411.04672, 2024.
21. Shao, Z. et al. Semantic-aware spectrum sharing in internet of vehicles based on deep reinforcement learning. *IEEE Internet Things J.* **23**(11), 38521–38536 (2024).
22. Vanneste, S., de Borrekens, G., Bosmans, S., et al. Learning to communicate with reinforcement learning for an adaptive traffic control system. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021)* 207–216 (Springer International Publishing, 2022)
23. Xie, H. et al. Deep learning enabled semantic communication systems. *IEEE Trans. Signal Process.* **69**, 2663–2675 (2021).
24. Li, A. et al. Cross-modal semantic communications. *IEEE Wirel. Commun.* **29**(6), 144–151 (2022).
25. Chen, M. et al. Cross-modal graph semantic communication assisted by generative AI in the metaverse for 6G. *Research (Washington DC)* **7**, 0342 (2024).
26. Zhao, H., Li, H., Xu, D., et al. Multi-Modal Self-Supervised Semantic Communication. arXiv preprint arXiv:2503.13940, (2025).
27. Wang, Y., Liao, J., Yu, H., et al. Advanced Conditional Variational Autoencoders (A-CVAE): Towards interpreting open-domain conversation generation via disentangling latent feature representation. arXiv preprint arXiv:2207.12696, (2022).
28. Zhou, Z. et al. Dynamic attention-based CVAE-GAN for pedestrian trajectory prediction. *IEEE Robot. Autom. Lett.* **8**(2), 704–711 (2022).
29. Fang, L., Zeng, T., Liu, C., et al. Transformer-based conditional variational autoencoder for controllable story generation. arXiv preprint arXiv:2101.00828, (2021).
30. Bao, F., Nie, S., Xue, K., et al. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning* 1692–1717. (PMLR, 2023)
31. Prakash, A., Chitta, K. & Geiger, A. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 7077–7087 (2021).
32. Maaz, M. et al. Class-agnostic object detection with multi-modal transformer. In *European Conference on Computer Vision* 512–531 (Springer, 2022).
33. Zhang, Z. et al. How to cache important contents for multi-modal service in dynamic networks: A DRL-based caching scheme. *IEEE Trans. Multimedia* **26**, 7372–7385 (2024).
34. Tu, L. & Yan, X. Multi-modal inter-domain routing based on multi-agent reinforcement learning on heterogeneous networks. In *2024 International Conference on Engineering and Emerging Technologies (ICEET)* 1–6 (IEEE, 2024).
35. Khalil, Y. H. & Mouftah, H. T. Exploiting multi-modal fusion for urban autonomous driving using latent deep reinforcement learning. *IEEE Trans. Veh. Technol.* **72**(3), 2921–2935 (2022).
36. Dhiman, G. et al. Multi-modal active learning with deep reinforcement learning for target feature extraction in multi-media image processing applications. *Multimedia Tools Appl.* **82**(4), 5343–5367 (2023).
37. Zhang, R., Wang, B., Zhang, J., Bian, Z., Feng, C. & Ozbay, K. When language and vision meet road safety: leveraging multimodal large language models for video-based traffic accident analysis. arXiv preprint arXiv:2501.10604.

Author contributions

Jiajun Zou: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation Zhiping Wan: methodology, software, validation Feng Wang: Conceptualization, methodology Shitong Ye: formal analysis, investigation, resources, data curation Shaojiang Liu: writing—review and editing, visualization, supervision, project administration, funding acquisition.

Funding

This research was supported by the Characteristic Innovation Category Project of Guangdong Ordinary Colleges and Universities (No. 2024KTSCX127), the Young Innovative Talents Category Project of Guangdong Ordinary Colleges and Universities (Nos. 2023KQNCX124, 2024KQNCX076), the 2024 Natural Science Platforms and Projects for General Colleges and Universities in Guangdong Province (No. 2024ZDZX3035), the School-level Scientific Research Project of Guangzhou Xinhua University (No. 2024KYCXTD02), and the Guangdong Province Key Construction Discipline Research Capacity Enhancement Project (Nos. 2021ZDJS144, 2024ZDJS130).

Declarations

Competing interests
The authors declare no competing interests.

Ethics statement
This article does not contain any studies with human participants or animals performed by any of the authors. All methods were performed in accordance with relevant guidelines and regulations.

Additional information
Correspondence and requests for materials should be addressed to S.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.