Heterogeneous Graph Transformer for Connected Autonomous Vehicles' Cooperative Localization

Shengsheng Xing, Haigen Min, *Senior Member, IEEE*, Xia Wu, Wuqi Wang, *Graduate Student Member, IEEE*, Xiangmo Zhao, *Member, IEEE*, and Guoqiang Mao, *Fellow, IEEE*

***Abstract*—Cooperative localization (CL) is a key enabler for future cooperative intelligent transportation system applications. By sharing and fusing multisource information, it can provide high-accuracy localization for connected autonomous vehicles in GNSS-denied or limited environments. Practical CL scenarios often involve heterogeneous cooperative nodes, such as vehicles and landmarks, and relative measurements (e.g., range and bearing). However, existing graph neural network (GNN)-based CL methods do not sufficiently account for scene heterogeneity or fully exploit relative information. Moreover, the state components estimated in CL exhibit substantial scale discrepancies, making effective joint optimization crucial for improving the overall estimation performance of the model. To address these issues, a heterogeneous CL Graph Transformer (HCLGT), together with an adaptive optimization strategy, is proposed for the real-vehicle CL task. First, we incorporate edge-based relative position encoding into the attention computation to better leverage relative measurements. Second, we design relation-specific aggregation modules for different measurement relationships and employ node uncertainty to adaptively adjust attention weights, improving robustness under heterogeneous information quality. Finally, an adaptive loss function is developed to dynamically balance the optimization of multiscale state components. The proposed method is validated on the real-world Leibniz University Cooperative Perception and Urban Navigation Dataset (LUCOOP). Experimental results show that, compared with the best-performing baseline, the proposed method reduces the localization RMSE by 44.61%. Extensive experimental analyses further validate the effectiveness of each module.**

***Index Terms*—Connected and autonomous vehicles (CAVs), cooperative localization (CL), deep learning, Graph Transformer.**

I. Introduction

CONNECTED and autonomous vehicles (CAVs) play a key role in improving the efficiency and safety of transportation systems [1]. Reliable self-localization is fundamental to the perception, planning, and control of CAVs [2]. However, mainstream localization technologies, such as GNSS, often suffer from performance degradation or even failure in complex environments, where adverse conditions, such as multipath, non-line-of-sight (NLOS) [3], and intrinsic sensor limitations can reduce localization accuracy [4], thereby leading to unreliable perception and unsafe decision-making in CAV systems [5]. Cooperative localization (CL) provides a promising solution to this problem. By leveraging the Internet of Vehicles and onboard perception, CAVs can share and fuse additional information from surrounding vehicles and infrastructure, thereby improving localization accuracy [6]. Fig. 1 illustrates a CL scenario.

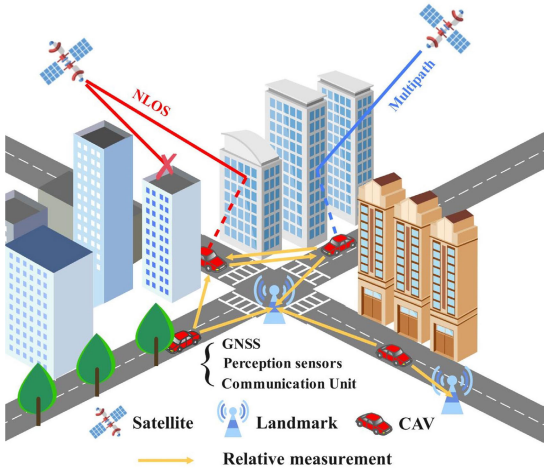


Fig. 1. Illustration of CL in a typical urban scenario.

Various algorithms have been developed for CL under different estimation paradigms, including the Kalman filter (KF) [7], factor graph optimization (FGO) [8], and belief propagation (BP) [9]. Among them, BP-based methods have received considerable attention because they represent the CL system as a graph and perform distributed information fusion [10]. However, BP may converge to suboptimal solutions when the graph contains loops or when the system model is nonlinear and non-Gaussian [11]. Graph neural networks (GNNs), with their ability to learn internode relationships and aggregate neighborhood information, provide an effective data-driven framework for graph-based CL. Existing GNN-based

Received 14 July 2026; accepted 30 July 2026. Date of publication 5 August 2026; date of current version 8 October 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 52372426, Grant 52302399, and Grant U25B20136; and in part by the Fundamental Research Funds for the Central Universities, Chang'an University under Grant 300102243202 and Grant 300119246501. *(Corresponding author: Haigen Min.)*

Shengsheng Xing, Haigen Min, Xia Wu, Wuqi Wang, and Xiangmo Zhao are with the School of Information Engineering, Chang'an University, Xi'an 710064, China, and also with the Collaborative Innovation Center for Western China Traffic Safety and Intelligent Cooperative Control, Xi'an 710064, China (e-mail: shengshengxing@chd.edu.cn; hgmin@chd.edu.cn; wuxia@chd.edu.cn; wuqiwang@chd.edu.cn; xmzhao@chd.edu.cn).

Guoqiang Mao is with the School of Transportation, Southeast University, Nanjing 211189, China (e-mail: 101013900@seu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2026.3720786

2327-4662 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

CL methods have shown improved localization accuracy and robustness over traditional approaches. Therefore, designing GNN models tailored to CL has become an important direction for further improving localization performance.

In CL, graph structures provide a natural framework for modeling interactions among connected vehicles and roadside units. Nodes carry attributes such as state and type, while edges represent relative measurement relationships between nodes. Through neighborhood aggregation and message passing, vehicle states can be updated and estimated. Although conventional GNNs have been widely studied, they still suffer from limited expressive power [12], oversmoothing [13], and oversquashing [14]. Graph Transformers [15], built upon the self-attention mechanism of Transformers [16], provide a powerful alternative for modeling complex node interactions. However, directly applying a standard Graph Transformer to CL is insufficient because its attention computation mainly relies on node representations and does not explicitly exploit edge-encoded relative measurements. This limitation is critical for CL, where edge features contain physically meaningful geometric constraints, such as distance and angle. Ignoring these measurements forces the model to infer internode relationships only from node states, increasing learning difficulty and weakening the representation of cooperative geometric relationships [17]. Therefore, incorporating edge information into attention computation is necessary for enabling Graph Transformers to better capture relative measurement constraints and improve localization accuracy.

Heterogeneity is an inherent property of CL systems. Real-world CL scenarios usually involve different types of cooperative nodes, such as vehicles and landmarks, and different measurement relations. However, many graph representation learning methods used for CL are still based on homogeneous graph assumptions or unified aggregation mechanisms [18]. Such designs may overlook feature differences among node types and oversimplify relation-specific measurement characteristics. This is insufficient for CL graphs with explicit physical meanings [19]. Specifically, vehicles and landmarks differ significantly in observation sources and information reliability. Existing CL studies [20], [21], [22] usually regard static anchors or landmarks as reference nodes with known positions, whereas neighboring vehicles are still associated with position uncertainty. Therefore, landmark information is generally more stable and has more controllable error statistics than information from neighboring vehicles. Moreover, different relation types contribute differently to target vehicle state updates. Fusing them in a unified manner may weaken the model's ability to represent complex cooperative relationships and degrade localization accuracy. Therefore, explicitly modeling node-type and relation-type differences, as well as their relative contributions during information fusion, remains a key issue for heterogeneous CL graphs.

Previous deep-learning-based CL methods typically formulate vehicle state estimation as a unified regression problem and optimize it using the mean squared error (mse) loss [23]. However, in real-world open-road driving scenarios, different vehicle state components often have markedly different dynamic ranges. In particular, vehicle motion is mainly reflected in horizontal position variations, whereas vertical displacement is usually much smaller and becomes evident only in scenarios such as overpasses, tunnel entrances and exits, or ramps. This scale mismatch makes it difficult for a unified regression loss to simultaneously optimize horizontal and vertical components [24]. Although using a single model to jointly predict multiple state components is beneficial for reducing onboard deployment cost, simply summing componentwise losses may cause training to be dominated by components with larger scales or higher noise levels. Manual normalization or heuristic weighting can partially alleviate this issue, but their weight settings usually rely on empirical choices. Therefore, for deep-learning-based CL models, adaptively balancing the optimization contributions of horizontal and vertical state components during training remains important for improving overall localization accuracy.

However, applying Graph Transformers to CL in real autonomous driving scenarios still faces multiple challenges. Specifically, the model should take into account the relative measurement constraints encoded in edge features during attention computation while also distinguishing heterogeneous nodes such as vehicles and landmarks and their associated measurement relations. In addition, the optimization of multiscale vehicle state components must be properly balanced during training. These issues jointly increase the difficulty of adapting and optimizing Graph Transformers for real-world CL scenarios. The main contributions of this study are given as follows.

1) By incorporating edge features into the dot product attention computation, a relative position encoding-enhanced attention mechanism is proposed. This design enables more accurate modeling of internode relationships in CL systems, thereby improving state estimation accuracy.
2) Building on the relative position encoding enhanced attention mechanism, a two-stage aggregation Graph Transformer module is further designed for heterogeneous CL. It can aggregate neighborhood information under the same measurement relation while adaptively characterizing the relative contributions of different relation types to target vehicle state updates.
3) An adaptive loss function for state estimation in CL systems is proposed. The method formulates the horizontal and vertical state components as two regression subtasks and introduces a homoscedastic uncertainty learning mechanism to adaptively adjust the loss weights of these subtasks, thereby improving the model's ability to jointly optimize horizontal and vertical localization components.

The remainder of this article is organized as follows. Section II reviews the related work. Section III presents the problem formulation and describes the proposed method in detail. Section IV describes the experimental results. Section V concludes our work.

II. Related Work

Classical CL methods are mainly built on traditional state estimation frameworks [3], [9], and their performance typically

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

depends heavily on prior assumptions about noise models, motion dynamics, and measurement distributions. By leveraging recent advances in deep learning, GNNs have opened up new opportunities for CL. Early studies introduced GNNs into CL to learn correction functions for conventional BP in complex scenarios, thereby compensating for its estimation errors [11]. Subsequent work further developed a two-stage CL framework, in which temporal priors are first obtained through a sequential model and cooperative fusion is then performed via graph-based message passing [25]. GNNs have also been used to learn compact graph representations to address the data association problem in cooperative systems [26]. For communication-constrained real-world deployment, VQ-MPNN [23] compresses node states during message passing into an acceptable communication budget through a learnable vector quantization codebook, further improving the practicality of GNN-based CL. Among these methods, MPNN is one of the most representative architectures, whose core idea is to progressively aggregate neighborhood information and update node representations through multilayer message passing. However, it still has a limited ability to capture more flexible dependence patterns among neighboring nodes.

Transformers have achieved remarkable success across a wide range of learning tasks, largely because the self-attention mechanism can adaptively model pairwise dependencies among elements [16]. Inspired by this, recent studies have begun to explore the potential of Graph Transformers in CL. Some methods enhance node representations through self-attention and combine them with graph convolution for position estimation [27]. Other methods introduce attention mechanisms to dynamically adjust the importance of neighboring nodes or edges under noisy observations, thereby improving model robustness [28]. However, standard self-attention does not explicitly exploit graph topology or edge attributes and, therefore, cannot fully capture the structural and geometric relationships encoded in graphs. Because driving scenes exhibit strong spatial characteristics, existing Transformer-based methods for driving scenarios often introduce geometric priors into attention computation through reference coordinate transformations or pairwise relative pose encoding. For example, Motion Transformer [29] provides explicit scene geometry for local self-attention through target-centered coordinate transformation and positional encoding. HPTR [30] constructs positional encodings based on the relative poses between a query polyline and its neighboring polylines, and injects them into the attention mechanism. In general, Graph Transformers, when edge features describe relative structural or geometric relationships between node pairs, can also be regarded as graph relative positional encoding (RPE) [31]. GRPE [32] jointly encodes node topological relationships and edge attributes to strengthen node–edge interactions. These studies indicate that enhancing attention with relative relationships and edge information has become an important direction in graph representation learning. However, there is still a lack of targeted research on how to effectively incorporate such relative measurements into heterogeneous Graph Transformers for CL.

General heterogeneous GNNs aim to learn unified representations from graphs containing multiple node types and relation types. Early heterogeneous GNNs typically relied on meta path-based neighborhood aggregation. For example, the heterogeneous graph attention network introduced a hierarchical attention mechanism [33]. Another representative line of work is the relational graph convolutional network, which distinguishes different edge types through relation-specific weight matrices and normalization strategies [34]. Subsequently, heterogeneous Graph Transformers further incorporated node type- and edge type-dependent projection transformations and relation-aware attention mechanisms within the Transformer framework [35]. These pioneering studies laid the foundation for general heterogeneous graph architectures capable of handling complex graph patterns. In traffic scenarios, scene entities and interaction relationships are inherently heterogeneous [36]. Early work such as VectorNet [37] used GNNs to process vectorized representations of map elements and traffic participant trajectories, but it did not explicitly characterize how different entity types and relation types affect interaction strength. Heterogeneous Graph Transformers typically model heterogeneous entities and interaction relationships in traffic scenes explicitly according to entity types, such as traffic participants and map elements, as well as the semantic relations among them. LaneGCN [38] separately models multiple categories of interactions, including those between traffic participants and lanes, within lanes, and among traffic participants. HEAT [39] leverages heterogeneous edges to enhance attention for modeling relationships among different traffic participants. SemanticFormer [40] uses a semantic traffic scene graph to model the semantic relations among road topology, traffic participants, and traffic elements. HDGT [19] further aggregates multitype relations through a hierarchical Transformer. However, its aggregation results from different relation types are mainly fused implicitly through concatenation and an MLP, without further explicitly estimating the relative importance of different relation types for the target node. These studies indicate that heterogeneous Graph Transformers have been widely used for trajectory prediction and scene understanding in driving scenarios. However, these methods are primarily designed for trajectory prediction and scene understanding, where heterogeneous relations are usually defined by traffic semantics, road topology, or scene interactions, whereas CL is more concerned with information fusion relationships formed by the relative measurement process. Real-world CL systems usually involve multiple types of cooperative agents. Therefore, a well-designed CL framework should account for different input types and their relationships. Recently, some studies have begun to consider heterogeneity in CL scenarios. One line of work distinguishes cooperative information according to its source, treating uncertain neighboring vehicles as virtual landmarks and calibrated landmarks as real landmarks [27]. Furthermore, some studies have investigated how to distinguish absolute measurements from known landmarks and the inherent uncertainty of vehicle nodes when updating states using relative measurements from neighbors and accordingly designed two position update mechanisms for landmarks or anchors and neighboring vehicles, respectively [22].

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

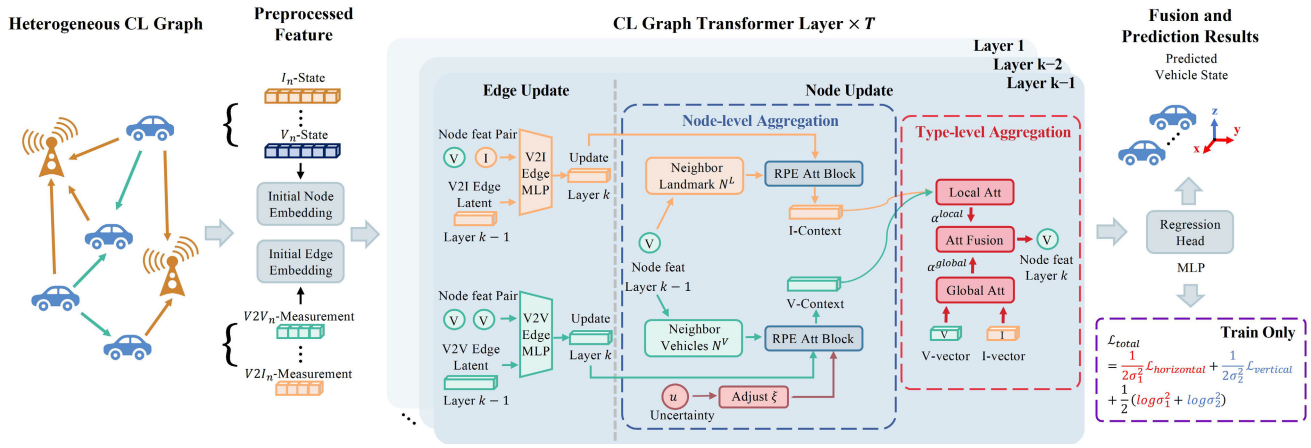


Fig. 2. Pipeline of HCLGT.

Another line of work classifies network nodes into base stations and user terminals and explicitly performs category-aware fusion through self-attention during aggregation [28], but it ignores the dynamic contributions of different relation categories to target node state updates.

Most existing deep learning-based CL methods are optimized using the mse loss [23]. When different state components exhibit similar scale characteristics, this design is simple and effective. However, in real driving scenarios, state components often differ substantially in their dynamic ranges. Existing studies have attempted to assign component weights via scale normalization [25], but such strategies often rely on prior estimates. Multitask learning (MTL) jointly models multiple tasks by sharing weights and computation across them and, thus, has the potential to improve task performance while reducing overall computational cost and learning better shared representations [41]. From this perspective, CL naturally involves the simultaneous estimation of multiple state variables and can, therefore, be formulated as an MTL problem [42]. In other tasks that involve the estimation of multiscale state components, prior studies have proposed adaptively balancing the substantial differences in units and scales across components using learnable weights, rather than relying on manually specified loss weights [43]. Further studies have shown that multitask loss weights are closely related to the measurement scales and uncertainty levels of individual tasks and are largely influenced by the magnitude of task noise [44]. In vehicle localization, jointly regressing 3-D translation and rotation with a single loss function has been shown to yield more reliable localization results [45].

III. Methodology

A. Problem Formulation and Approach Overview

Consider the CL scenario for CAVs. All CAVs are equipped with navigation devices and can obtain their own state information. In addition, each CAV carries onboard sensors to acquire relative measurements of surrounding vehicles and landmarks. When vehicle i detects vehicle (or landmark) j, a communication link is established to share state information. The problem studied in this article is defined as follows: we construct a CL system based on relative measurements. Given the state information of vehicles and landmarks, together with the relative measurements between vehicles and vehicles (or landmarks), our goal is to jointly estimate the current states of the vehicles.

The pipeline of Heterogeneous CL Graph Transformer (HCLGT) is shown in Fig. 2, which involves the following.

1) Section III-B describes how to construct a heterogeneous graph for the CL scenario from the raw inputs.
2) Section III-C introduces how the Graph Transformer architecture is used to aggregate and update node and edge features.
3) Section III-D presents the regression head designed based on the updated vehicle node features, together with the loss function used.

B. Construction of a Heterogeneous CL Graph

In this section, we represent the CL scenario as a directed heterogeneous graph based on relative measurements. The directed graph is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. A heterogeneous graph is associated with a node-type set $\mathcal{A} = \{V, L\}$ and an edge-type set $\mathcal{R} = \{V \to V, V \to L\}$, where V denotes vehicles and L denotes landmarks, with a node-type mapping function $\tau(v) \to \mathcal{A}$ and an edge-type mapping function $\varphi(e) \to \mathcal{R}$. An edge $e_{i \to j}$ with $i \neq j$ indicates that a relative measurement exists from the source node i to the target node j. It is worth noting that, in practical applications, due to heterogeneous sensor configurations, measurement failures, and field of view occlusions, relative measurements between nodes are usually directional and asymmetric. In other words, the fact that node i can observe node j does not imply that node j can simultaneously observe node i. Therefore, edges are modeled as directed. Moreover, as measurement availability changes over time, V2V and V2I observations are intermittent, and the graph topology also varies dynamically. Hence, the constructed heterogeneous CL graph is both directed and dynamic.

Next, we initialize feature embeddings for all nodes and edges. The feature vector of the i node in the scene can be denoted as $f_i := [x_i, y_i, z_i, c_i]^\top$, where $[x_i, y_i, z_i]^\top$ is its spatial

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

position in a 3-D coordinate system and c_i denotes the one hot encoding of the node i category. Our method can be naturally extended to CL systems with more node categories. We then generate the initial node features $v_i^0 = \mathrm{MLP}_{\tau(v_i)}(f_i)$. Following the underlying physical structure, we represent the edge feature of e_{ij} using relative measurements. The edge encoding can be formulated as $e_{i\to j}^0 = \mathrm{MLP}_{\varphi(e_{i\to j})}(z_{ij})$ for $j \in \mathcal{N}_i$. Here, $\mathcal{N}_i$ denotes the set of nodes measured by vehicle i, and the actual measurement is z_{ij}, which typically includes distance and angular information.

In summary, this section models the CL system as a directed heterogeneous graph based on relative measurements. We initialize node feature embeddings based on the motion/attribute characteristics of vehicles (or landmarks) and edge feature embeddings using relative measurements from vehicles to other vehicles (or landmarks).

C. CL Graph Transformer Layer

The next step is to perform information exchange between nodes and edges via aggregation and update functions and to update their features. The aggregation and update schemes for nodes and edges are presented in Sections III-C1 and III-C4, respectively.

1) Node Aggregation and Update Functions: In heterogeneous CL systems, nodes from different categories interact in different ways. Therefore, interactions of different types should be modeled separately, and different categories do not contribute equally to the target node. Moreover, even within the same category, different neighbors may have different levels of influence on the target node. To this end, we propose a novel two-stage aggregation Transformer that fuses information from heterogeneous neighbors, comprising node- and type-level aggregations. The CL Graph Transformer layer performs T rounds of node and edge updates and returns the updated vehicle-node features at the final iteration.

2) Node-Level Aggregation: The node-level aggregation module aggregates information from a target node and all its neighbors of a specific category. For each relation type, the model assigns a corresponding aggregation module.

For the vehicle–landmark aggregation function, we formally denote the landmark neighborhood of a target vehicle node v_i^{t-1} at layer $t-1$ as $\mathcal{N}_i^L = \{v_j^{t-1}\}_{j=1}^m$. It should be noted that directly fusing states represented in global coordinates often makes the model less robust to translation and rotation transformations [30]. To address this issue, we need to explicitly exploit the relative geometric information encoded in the edge features. Therefore, when updating v_i^{t-1}, it is necessary to consider both the features of neighboring nodes and the corresponding edge features. Motivated by this, we propose cross-attention with relative position encoding as the aggregation function.

Specifically, we take the target-node feature v_i^{t-1} as *Query* and use the neighbor node features as *Key* and *Value*. First, we need to obtain the RPE. Directly using the high-dimensional edge latent $e_{i\to j}^{t-1}$ as the relative-geometry input can lead to numerical scale instability, thereby weakening cross-scenario generalization. To address this, we regress the updated edge feature $e_{i\to j}^{t-1}$ to a low-dimensional relative geometric representation $\hat{r}_{ij}^{t-1} = (\hat{d}_{ij}^{t-1}, \hat{\theta}_{ij}^{t-1})$, where $\hat{d}_{ij}^{t-1}$ and θ_{ij}^{t-1} denote the distance and bearing angle, respectively. Considering the physical constraints that distances are nonnegative and bearing angles are bounded, we impose explicit constraints on this low-dimensional output, specifically

$$\hat{d}_{ij}^{t-1} = \mathrm{softplus}\left(e_{i\to j}^{t-1}\right), \quad \hat{\theta}_{ij}^{t-1} = \pi \tanh\left(e_{i\to j}^{t-1}\right). \tag{1}$$

This ensures that $\hat{d}_{ij}^{t-1} \geq 0$ and $\hat{\theta}_{ij}^{t-1} \in [-\pi, \pi]$. Then, to better capture high-frequency variations in the data [46], we apply sinusoidal positional encoding to $\hat{d}_{ij}$ and sinusoidal angular encoding [47] to $\hat{\theta}_{ij}$, respectively,

$$\mathrm{RPE}\left(\hat{r}_{ij}^{t-1}\right) = \mathrm{concat}\left(\mathrm{PE}\left(\hat{d}_{ij}^{t-1}\right), \quad \mathrm{AE}\left(\hat{\theta}_{ij}^{t-1}\right)\right) \tag{2}$$

where

$$\mathrm{PE}_{2i}(x) = \sin\left(x \cdot \omega^{2i/D}\right), \quad \mathrm{PE}_{2i+1}(x) = \cos\left(x \cdot \omega^{2i/D}\right) \tag{3}$$

$$\mathrm{AE}_{2i}(x) = \sin\left(x(i+1)\right), \quad \mathrm{AE}_{2i+1}(x) = \cos\left(x(i+1)\right) \tag{4}$$

where ω denotes the base frequency and D is the encoding dimension, $i \in \left\{0, 1, \ldots, \frac{D}{2} - 1\right\}$. Moreover, this design naturally extends to 3-D relative-measurement settings that include a pitch angle. When a pitch angle is introduced, we adopt the same regression constraint and angular encoding strategy as used for the bearing angle, thereby improving the scalability of our method. Then, the RPE can be incorporated into the *Query*, *Key*, and *Value* [48]. The RPE block is illustrated in Fig. 3. Our experiments show that injecting RPE into *Query* does not improve performance. Therefore, we add RPE to the *Key* and *Value* in the attention computation

$$e_{ij}^{t-1,L} = \frac{\left(v_i^{t-1} W_L^{Query}\right)\left(v_j^{t-1} W_L^{Key} + \mathrm{RPE}\left(\hat{r}_{ij}^{t-1}\right)\right)^{\mathsf{T}}}{\sqrt{h}} \tag{5}$$

$$\alpha_{ij}^{t-1,L} = \frac{\exp\left(e_{ij}^{t-1,L}\right)}{\sum_{l\in\mathcal{N}_i^L} \exp\left(e_{il}^{t-1,L}\right)} \tag{6}$$

where W_L^Q and W_L^K are the linear projection matrices for the *Query* and *Key* of the vehicle–landmark relation and $\alpha_{ij}^{t-1,L}$ denotes the attention coefficient between vehicle node i and landmark node j under this relation at layer $t-1$. Accordingly, the aggregated feature of node i under this relation is obtained as $v_i^{t-1,L}$

$$v_i^{t-1,L} = \sum_{l\in\mathcal{N}_i^L} \alpha_{il}^{t-1,L}\left(v_l^{t-1} W_L^{Value} + \mathrm{RPE}\left(\hat{r}_{il}^{t-1}\right)\right). \tag{7}$$

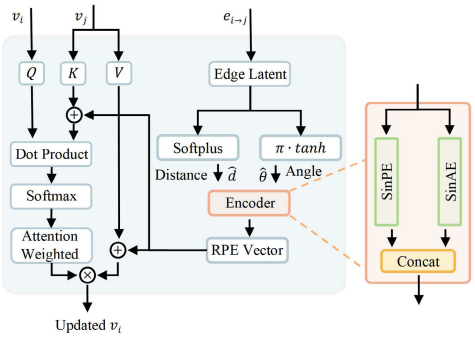


Fig. 3. Illustration of the proposed RPE block.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

For the vehicle–vehicle aggregation function, uncertainty in vehicle-node positions should be explicitly considered during message exchange. This is because aggregating low-quality information from neighbors can disturb the target node's embedding learning, leading to suboptimal performance [49]. Therefore, when computing attention weights for vehicle–vehicle aggregation, we leverage the uncertainty of vehicle states to characterize the information quality of neighboring vehicles and adaptively downweight directed edges from low-quality vehicle neighbors. The trace of a covariance matrix is commonly used as a scalar uncertainty measure. For a vehicle, the trace of its position covariance matrix reflects the overall dispersion of the estimated position across coordinate dimensions and can serve as an indicator of state accuracy [50]. A vehicle node with a large trace value implies substantial uncertainty and may propagate errors during information fusion. Given the covariance matrix $P_j^{t-1} \in \mathbb{R}^{d_p \times d_p}$ of vehicle node j, which is typically provided by the initial localization module, where d_p denotes the dimensionality of the vehicle position, its uncertainty is defined as

$$u_j^{t-1} = \text{trace}\left(P_j^{t-1}\right). \tag{8}$$

We then dynamically adjust the influence of each neighboring vehicle on the target node according to the neighbor's uncertainty. Specifically, for highly uncertain neighbors, we restrict their feature propagation to the target node. Meanwhile, we strengthen connections (i.e., assign higher weights) to low-uncertainty neighbors so that the target node receives more reliable information. To this end, we adopt an uncertainty-aware scheme that guides attention weight adjustment based on neighbor uncertainty. Formally, given the uncertainty measure of node j, the adjustment coefficient ξ_j^{t-1} is defined as

$$\xi_j^{t-1} = \psi\left(e^{-u_j^{t-1}} - \varepsilon\right) \tag{9}$$

where ε is a learnable parameter that serves as a threshold to distinguish high-uncertainty and low-uncertainty vehicle nodes. $e^{-u_j^{t-1}}$ normalizes the node uncertainty into the range $[0, 1]$, and the function $\psi(\cdot)$ is defined as

$$\psi(x) = \begin{cases} \alpha \cdot s(x), & x \geq 0 \\ \beta, & x < 0 \end{cases} \tag{10}$$

where $s(x)$ is the sigmoid function, $\alpha > 0$ is a hyperparameter used to amplify edge weights from low-uncertainty neighbors, and $\beta > 0$ is a hyperparameter that controls the degree of downweighting for edges from high-uncertainty neighbors. Intuitively, when $e^{-u_j^{t-1}} < \varepsilon$, node j exhibits higher uncertainty and, thus, receives a smaller weight during information propagation. When $e^{-u_j^{t-1}} \geq \varepsilon$, we assign greater attention to such high-quality vehicle nodes.

After obtaining the adjustment coefficients for neighboring vehicles, we formulate an uncertainty-aware vehicle–vehicle aggregation function as follows:

$$e_{ij}^{t-1,V} = \left(\frac{\left(v_i^{t-1} W_V^{Query}\right)\left(v_j^{t-1} W_V^{Key} + \text{RPE}\left(\hat{r}_{ij}^{t-1}\right)\right)^{\top}}{\sqrt{h}} \right) \xi_j^{t-1} \tag{11}$$

$$\alpha_{ij}^{t-1,V} = \frac{\exp\left(e_{ij}^{t-1,V}\right)}{\sum_{l \in \mathcal{N}_i^V} \exp\left(e_{il}^{t-1,V}\right)} \tag{12}$$

where W_V^Q and W_V^K are the linear projection matrices for the *Query* and *Key* of the vehicle–vehicle relation, and $\mathcal{N}_i^V$ denotes the vehicle neighboring. $\alpha_{ij}^{t-1,V}$ is the attention coefficient between vehicle node i and vehicle node j under this relation, which is adjusted by the uncertainty-aware factor ξ_j and then normalized. Accordingly, the aggregated feature of node i under this relation is obtained as $v_i^{t-1,V}$

$$v_i^{t-1,V} = \sum_{l \in \mathcal{N}_i^V} \alpha_{il}^{t-1,V}\left(v_l^{t-1} W_V^{Value} + \text{RPE}\left(\hat{r}_{il}^{t-1}\right)\right). \tag{13}$$

By applying the above aggregation to each relation type, we obtain a set of relation-specific aggregated features $\{v_i^{t-1,n} \mid n \in \mathcal{R}\}$, which are then fused in the subsequent type-level aggregation module.

3) Type-Level Aggregation: The type-level aggregation module is designed to further fuse the relation-specific information that a target node obtains from node-level aggregation across different neighbor types. A straightforward strategy is to sum or concatenate these representations, but this implicitly assumes that all relation types contribute equally to the target node. A more reasonable design should satisfy two requirements simultaneously. First, the importance of a relation type can exhibit local heterogeneity across target nodes, meaning that different nodes may assign different weights to the same relation [51]. Second, the importance of different relations often follows certain global regularities (i.e., general patterns) that should be learned from a global perspective [52]. Therefore, our type-level aggregation jointly considers both local and global type importance.

For local type importance, we first compute the importance of each relation type to the target node i

$$\text{local}_i^{t-1,n} = \frac{\exp\left(\text{LeakyReLU}\left(q_{\text{local}}^{\top}\left[v_i^{t-1} \,\|\, v_i^{t-1,n}\right]\right)\right)}{\sum_{m \in \mathcal{R}} \exp\left(\text{LeakyReLU}\left(q_{\text{local}}^{\top}\left[v_i^{t-1} \,\|\, v_i^{t-1,m}\right]\right)\right)} \tag{14}$$

where q_{local} is a learnable attention vector and $\text{local}_i^{t-1,n}$ denotes the normalized local importance score of type n for node i.

For global type importance, to prevent the model from overrelying on local neighborhoods—which may lead to local optima—and to mitigate interference introduced by noisy connections, we introduce type importance from a global perspective. This term reflects the overall importance of a given type across all nodes and does not depend on any specific node features. Specifically, the global importance of type n is defined as

$$\text{global}_i^{t-1,n} = \frac{\exp\left(\text{LeakyReLU}\left(q_{\text{global}}^{\top} h_n\right)\right)}{\sum_{m \in \mathcal{R}} \exp\left(\text{LeakyReLU}\left(q_{\text{global}}^{\top} h_m\right)\right)} \tag{15}$$

where q_{global} denotes learnable global type-attention parameters and h_n is a learnable embedding vector for type n. The next step is to jointly consider the local and global importance of each type for node i and fuse the node-level aggregated

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

information from different relation types to obtain the final aggregated node feature

$$\hat{v}_i^{t-1} = \sum_{n\in\mathcal{R}} \left(\lambda \text{local}_i^{t-1,n} + (1-\lambda)\,\text{global}_i^{t-1,n}\right) v_i^{t-1,n} \tag{16}$$

where λ is a learnable weight used to balance local and global importance.

At this point, node aggregation is completed. Next, we fuse each node's own representation with the aggregated neighborhood information through an update function and update the node embeddings as follows:

$$v_i^t = \phi_v\left(\text{concat}\left(v_i^0, v_i^{t-1}, \hat{v}_i^{t-1}\right)\right) \tag{17}$$

where $\phi(\cdot)$ denotes a feed-forward network. Note that although not explicitly shown in the equation, we also apply residual connections and layer normalization.

4) Edge Aggregation and Update Functions: Recall that the edge feature $e_{i\to j}$ represents the relative geometric information between nodes i and j from the perspective of node i. Therefore, updating $e_{i\to j}$ should jointly consider the features of both nodes i and j. Formally, for an edge feature $e_{i\to j}^t$ at layer t, we update it by aggregating the layer-t node features v_i^{t-1} and v_j^{t-1} and the edge feature $e_{i\to j}^{t-1}$.

For edge aggregation and update, considering that CL graphs contain different edge categories, we likewise adopt heterogeneity-aware edge aggregation and update functions

$$e_{i\to j}^t = \text{MLP}_{\varphi(e_{i\to j})}\left(\text{concat}\left(v_i^{t-1}, v_j^{t-1}, e_{i\to j}^{t-1}\right)\right) \tag{18}$$

where $e_{i\to j}^t$ is the updated feature.

D. State Inference and Training Strategy

After T message-passing steps, the CL Graph Transformer layer outputs the final updated node features v^T. We then use v^T to estimate the vehicle states

$$\hat{f} = \text{reg}\left(v^{\mathsf{T}}\right) \tag{19}$$

where reg$(\cdot)$ is implemented as an MLP regressor. It outputs a tensor of shape S, where S is the length of the state to be estimated. In this article, we set $S = 3$ corresponding to 3-D position estimation. Note that the output state dimension can be adjusted to meet different requirements. For example, depending on whether the state includes 2-D or 3-D position, different settings can be adopted to support diverse estimation needs.

Vehicle state estimation is a regression task, and it is typically optimized using an mse loss. However, different state components can differ substantially in scale ranges and noise statistics. To address this issue, we formulate CL state estimation as a multitask regression problem. Specifically, although all state variables are produced by a shared network, the estimation of horizontal position $[x, y]^{\mathsf{T}}$ and vertical position z is treated as two separate regression tasks

$$L_{horizontal} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(x_i^* - \hat{x}_i\right)^2 + \left(y_i^* - \hat{y}_i\right)^2\right] \tag{20}$$

$$L_{\text{vertical}} = \frac{1}{N}\sum_{i=1}^{N}\left(z_i^* - \hat{z}_i\right)^2 \tag{21}$$

where N denotes the number of vehicles to be estimated and $[x_i^*, y_i^*, z_i^*]^{\mathsf{T}}$ denotes the ground-truth spatial position of vehicle i. A central challenge in MTL is that different tasks may have different loss magnitudes, noise levels, and convergence rates. Naively summing them can cause one task to dominate training. To mitigate this issue, we adopt homoscedastic uncertainty to enable adaptive loss weighting [44], allowing the model to dynamically balance the contributions of different tasks during training. Specifically, we introduce dynamic noise factors σ that adjust the weights of subtask losses based on the data distribution of the training samples. Our final loss function is formulated as follows:

$$L_{\text{total}} = \frac{1}{2\sigma_1^2}L_{\text{horizontal}} + \frac{1}{2\sigma_2^2}L_{\text{vertical}} + \frac{1}{2}\left(\log\sigma_1^2 + \log\sigma_2^2\right) \tag{22}$$

where $(1/2)(\log\sigma_1^2 + \log\sigma_2^2)$ acts as a regularization term to mitigate overfitting, and σ_1 and σ_2 are learnable parameters.

During training, we adopt a centralized training paradigm, while, at inference time, a common set of learned parameters is used for heterogeneous CL graphs with different numbers of vehicles and landmarks. Benefiting from the parameter-sharing design, the proposed method can be directly applied to the CL scenario with varying measurement topologies and different numbers of vehicles and landmarks, without changing the model architecture.

IV. Experiments

A. Experimental Setup and Description

1) Dataset Details: We evaluate our method on the widely used large-scale Leibniz University Cooperative Perception and Urban Navigation Dataset (LUCOOP) [53]. It provides time-synchronized multimodal data collected from three interacting measurement vehicles in a real-world urban environment. The test area is characterized by narrow single-lane streets and surrounding buildings with heights of approximately 15–25 m, forming a typical urban street–canyon scenario that is challenging for vehicle localization.

The three vehicles repeatedly drove along an inner-city triangular route in a platooning manner, together with onboard and roadside sensor deployments to emulate continuous CL. Each vehicle is equipped with a GNSS receiver, a LiDAR sensor, and an inertial measurement unit (IMU). In addition, UWB devices are deployed on vehicles and at static roadside sites to support both V2V and V2I measurements. A total station is used on selected road segments to provide high-precision reference measurements.

The dataset spans over 4 km and more than 26 min of cooperative driving. It includes over 2.5 h of 10-Hz raw GNSS-only observations, which are used as the noisy initial vehicle position information. In addition, it provides the square of the diagonal position covariance elements corresponding to the east, north, and height coordinates, from which the vehicle position covariance matrix in the ENU coordinate system can be obtained. The dataset also contains 1874 V2V and 267 V2I UWB measurements, which serve as relative observation information. Each UWB record is accompanied by a range status indicator, where 0 denotes a successful

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

measurement and 1 denotes a failed measurement. Moreover, LUCOOP provides 10-Hz ground-truth trajectories for all three vehicles over the entire sequence, while the positions of the static UWB units are measured by a total station and, thus, have high accuracy. GPS time is used as the unified time reference for synchronization.

2) Metrics: We use mean absolute trajectory error (meanATE), root mean square error of absolute trajectory error (rmseATE), and 95th-percentile position error (p95) as evaluation metrics. Formally, let the trajectory estimated by the model be denoted as $(\hat{f}^1, \hat{f}^2, \ldots, \hat{f}^K)$ where $\hat{f}^k = (\hat{x}^k, \hat{y}^k, \hat{z}^k)$, which denotes the estimated position of all vehicles in the CL system at time step k. Denote the ground-truth trajectories of all vehicles over K time steps as $(f^{*,1}, f^{*,2}, \ldots, f^{*,K})$.

meanATE measures the deviation of the estimated trajectory from the ground truth in Euclidean space and reflects the average quality of the estimation. The detailed formulation is given as follows:

$$\text{meanATE} = \frac{1}{K}\sum_{k=1}^{K}\left\|\hat{f}^k - f^{*,k}\right\|_2. \tag{23}$$

rmseATE places greater emphasis on the reliability with which the estimated trajectory remains aligned with the ground-truth trajectory, and a smaller rmseATE indicates more stable estimation results. The detailed formulation is given as follows:

$$\text{rmseATE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left\|\hat{f}^k - f^{*,k}\right\|_2^2} \tag{24}$$

p95 measures the Euclidean position error threshold below which 95% of the errors fall. It emphasizes the worst case performance bound of the system in complex scenarios and reflects the reliable and practically usable range of the estimation results. The detailed formulation is given as follows:

$$\text{p95} = \text{Quantile}_{0.95}\left(\left\{\left\|\hat{f}^k - f^{*,k}\right\|_2\right\}_{k=1}^{K}\right). \tag{25}$$

3) Implementation Details: The model is implemented in PyTorch, with graph construction and minibatch processing carried out using PyTorch Geometric. We use the LUCOOP dataset and split it into training, validation, and test sets according to mutually nonoverlapping continuous segments, with ratios of 70%, 20%, and 10%, respectively. The high-precision reference trajectories provided by the dataset are never used as model inputs during either training or inference. They are used only as supervision signals during training and for quantitative evaluation during testing.

For the graph $\mathcal{G}_k$ at time step k, all nodes are initialized according to their types. If a node is a vehicle node, the noisy GNSS observation at the corresponding timestamp is used as its spatial position feature, which is concatenated with a one-hot node type encoding and then input into the encoder MLP_V. If a node is a landmark node, the high-precision static UWB coordinates provided by the dataset are used as fixed spatial position features and are likewise concatenated with a one-hot node type encoding before being fed into the encoder MLP_L. For directed edge construction, following the graph construction rules in Section III-B, we use the range status indicator of each UWB observation as the criterion and create a directed edge between the corresponding nodes only when the associated UWB measurement is valid, while a failed UWB measurement is represented by the absence of the corresponding edge. In addition, each training sample corresponds to a timestamp-level graph, and each minibatch is formed by multiple graphs with different edge sets, thereby naturally supporting a dynamic heterogeneous graph structure.

For vehicle node uncertainty, we first use the square of the covariance elements corresponding to the east, north, and height coordinates and compute the trace of the position covariance matrix according to (8). To eliminate the effect of scale and improve numerical stability, we further apply a logarithmic transformation to the trace value, followed by min–max normalization based on statistics computed from the training set.

The hidden dimension is set to 16. We train the model using the Adam optimizer with an initial learning rate of 10^{-4}, for 1200 epochs, and a batch size of 64. The learning rate schedule follows an integrated warm-up plus cosine strategy: during the first 40% of the epochs, the learning rate is linearly warmed up from 10^{-6} to 10^{-4}. It is then decayed to 0 using cosine annealing. All experiments are conducted on a desktop PC equipped with an Intel Core i7-12700F CPU and an NVIDIA GeForce RTX 3070 GPU.

We briefly introduce the methods we compare with on the public dataset LUCOOP.

1) EKF, a traditional GNSS single-point positioning solution where each vehicle independently estimates its state using the extended KF.
2) VBRDCL [54], a robust distributed CL method, models the measurement noise using a Student's t distribution and employs a variational Bayesian framework to infer the joint posterior of vehicle states and noise statistics.
3) GraphSAGE [55] aggregates features from neighboring nodes on the communication graph to update the representation of each vehicle. Its key idea is to fuse each vehicle's own measurements with neighborhood information through a shared aggregation function.
4) Hetero-GNN [28], a heterogeneity-aware GNN baseline. It employs type-specific node encoders, together with relation-specific edge update and attention modules, for different measurement links.
5) MPNN [25] performs multiround message passing on the communication graph via an MPNN to fuse self-state information and relative measurements.
6) Graph Transformer [56] uses attention to fuse self-state information with relative measurements from neighboring nodes, thereby enabling adaptive neighborhood information aggregation.
7) GAT [15] introduces an attention mechanism into neighborhood message passing in GNNs, enabling the model to assign adaptive weights to different neighbors.

B. Overall Performance Evaluation

Table I and Fig. 4 show that, in terms of meanATE, all GNN-based methods significantly outperform the conventional

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

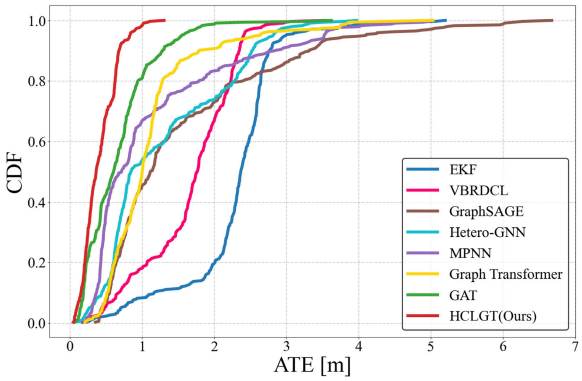


Fig. 4. CDFs of the L2 position error for all compared methods.

TABLE I
RESULTS OF THE OVERALL PERFORMANCE EVALUATION

Methods	meanATE↓	rmseATE↓	p95 ↓
EKF	2.2781	2.3794	2.9880
VBRDCL	1.6681	1.7786	2.4057
GraphSAGE	1.5693	1.9926	4.0704
Hetero-GNN	1.2802	1.5360	2.8115
MPNN	1.1357	1.5293	3.5215
Graph Transformer	1.1231	1.3238	2.6083
GAT	0.6819	0.8220	1.5934
HCLGT(Ours)	**0.3777**	**0.4263**	**0.7880**

fusion methods EKF and VBRDCL. This indicates that, in CL, end-to-end information fusion is more effective in reducing the overall average error. In addition, compared with the original EKF, VBRDCL exhibits stronger robustness in uncertain environments.

Further comparison among different GNN architectures shows that explicitly modeling graph interactions is important for CL. After incorporating edge features into message passing, MPNN achieves higher overall accuracy than GraphSAGE, suggesting that relative measurements are useful for cooperative information fusion. However, the p95 values of GraphSAGE and MPNN are even worse than those of some non-data-driven methods, indicating that these methods still suffer from noticeable degradation on a small number of difficult samples. This pattern can also be observed from the right tail of the cdf curves. Compared with GraphSAGE, Hetero-GNN achieves better performance on all three metrics, which demonstrates the benefit of incorporating heterogeneity into CL graph modeling. However, its meanATE and rmseATE are still inferior to those of MPNN and Graph Transformer. This suggests that simply introducing type-specific and relation-specific parameters is not sufficient for achieving optimal cooperative fusion. Compared to traditional aggregation functions, Graph Transformer employs attention-based aggregation to better capture dependencies and mitigate the large error tail problem. By introducing an attention mechanism, GAT can adaptively weight neighbor information, thereby substantially improving robustness to noise and low-quality observations. Accordingly, its cdf curve shifts left overall and exhibits a shorter right tail, indicating a clear reduction in the probability of large errors. Finally, the proposed HCLGT significantly

TABLE II
RESULTS OF MODEL COMPONENT ABLATION STUDY. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Heterogeneous	RPE	MTL	meanATE ↓	rmseATE ↓	p95 ↓
			1.9566	2.5171	5.3169
✓			1.4093	1.7399	3.1805
✓	✓		1.1763	1.3881	2.8302
✓	✓	✓	**0.3777**	**0.4263**	**0.7880**

outperforms all compared methods across all three metrics, demonstrating the importance of modeling heterogeneity and incorporating relative positional relationships into the attention computation as in HCLGT. Moreover, combined with the multitask loss design that balances scale discrepancies across different state components, the model achieves better overall performance.

In addition, the trajectory visualizations in Fig. 5 provide more intuitive evidence. Fig. 5(a) shows that HCLGT achieves the closest trajectory alignment with the ground truth on the real urban scene map, indicating its ability to suppress error accumulation and long-term drift. Fig. 5(b) further illustrates the height estimation results. When the raw position contains large vertical deviations, the baseline methods often fail to correct them promptly, whereas HCLGT can bring the height estimate back to a reasonable range. Fig. 5(c) shows the performance under different cooperation modes. HCLGT maintains smaller deviations in both V2V-only segments and segments with additional V2I constraints. These results further validate the effectiveness of modeling different measurement relations as a heterogeneous graph and performing relation-specific aggregation and fusion.

C. Ablation Study

We conduct the ablation study to validate the effectiveness of each design in HCLGT, and the results are reported in Table II. *RPE* indicates that edge features are first regressed into low-dimensional relative geometric relationships and then incorporated into the attention mechanism through sinusoidal encoding. Otherwise, edge features are not incorporated into the attention and are instead used directly for node updates. *Heterogeneous* indicates that different parameters are used for different types of nodes and edges, together with two-stage heterogeneous attention. Otherwise, the same parameters are shared, and a unified attention mechanism is applied to all neighboring nodes. *MTL* indicates that the problem is formulated as a multitask regression problem and optimized with an adaptive loss function based on homoscedastic uncertainty. Otherwise, it is treated as a single regression task and optimized using the mse loss. From the results, we can conclude the following.

1) *Heterogeneous Graph Design Is Crucial:* Assigning dedicated parameters to different types of nodes and edges, together with the proposed two-stage aggregation Transformer, significantly improves the model performance.
2) *Information Carried by Edge Features Is Beneficial:* After injecting RPE into the attention computation,

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

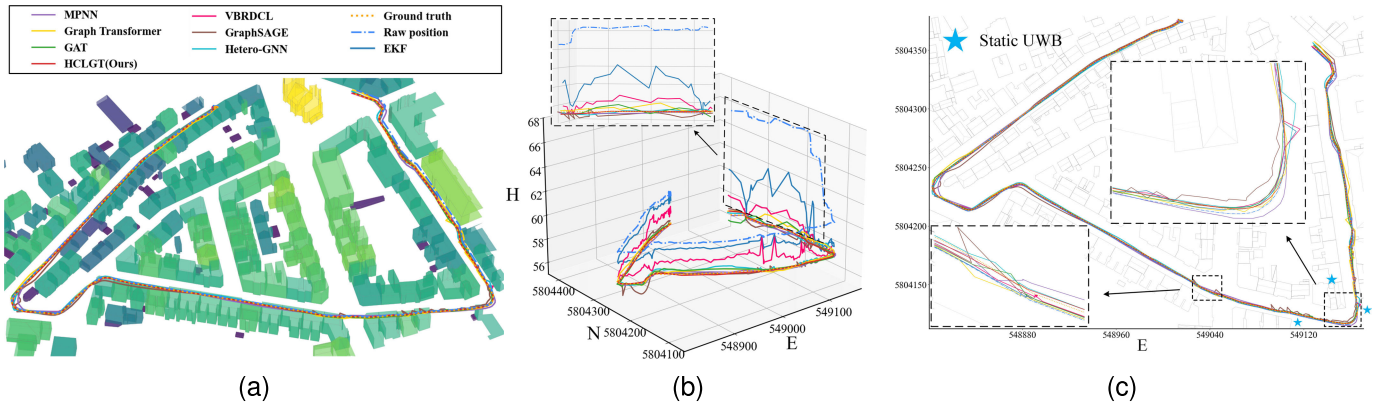


Fig. 5. Visualization of the estimation results of HCLGT and the compared methods on the LUCOOP open dataset. (a) Global trajectory and scene visualization. (b) 3-D trajectory comparison. (c) 2-D trajectory comparison.

the model achieves better performance, which confirms the effectiveness of edge feature information for both neighbor selection and message aggregation.

3) *Adaptive Loss Function Is Essential:* The full version of HCLGT achieves the best performance, and the adaptive multitask loss effectively alleviates the typical seesaw effect among different state estimation components, thereby significantly improving the overall model performance.

D. Analysis of RPE Block Design

In this section, we examine the following design choice.

1) *Direct Edge Encoding:* We directly encode the updated edge features as relative geometric inputs, in order to examine the necessity of using sinusoidal positional and angular encoding.
2) *Direct Latent Projection:* We directly project the updated high-dimensional edge latent features into RPE through an MLP, in order to examine the advantage of using low-dimensional geometric quantities regressed from edge latent features as inputs.
3) *MLP Function:* We remove the physical constraints imposed on the explicitly regressed edge measurements. Specifically, the predicted range and bearing are taken directly from the MLP outputs, without enforcing non-negativity or angular range limits, in order to examine the necessity of introducing physical prior constraints into explicit geometric measurement prediction.
4) *RPE in Attention:* We conduct an ablation study on different ways of incorporating RPE into dot product attention.

The results are in Table III, and we can conclude the following.

1) Directly using the updated edge features as relative geometric inputs causes the largest performance degradation. This is consistent with our expectation because relative measurements in real environments often exhibit stronger high-frequency variations and, therefore, require appropriate inductive bias.
2) Directly projecting high-dimensional edge features into RPE vectors leads to a degradation in performance. This suggests that directly using high-dimensional edge latent features is insufficient to provide effective geometric guidance for CL. A possible reason is that regressing low-dimensional geometric quantities enables the model to preserve physically meaningful relative measurement information before constructing the RPE.
3) Directly projecting high-dimensional edge latent features into RPE leads to performance degradation. In contrast, regressing low-dimensional geometric quantities helps preserve physically meaningful relative measurement information before constructing the RPE.
4) The results show that incorporating RPE into k and v brings more evident gains. A reasonable explanation is that, in dot product attention, k directly determines the matching scores and neighbor selection, and injecting RPE into k, therefore, enables the model to favor more relevant cooperative edges. Adding RPE to v, in turn, provides relative geometric context for the aggregated information.

TABLE III
RESULTS OF RPE BLOCK DESIGN

Models	meanATE↓	rmseATE↓	p95 ↓
Direct Edge Encoding	0.5670	0.7151	1.3107
Direct Latent Projection	0.4320	0.5328	1.1074
MLP Function	0.4516	0.5154	0.9331
RPE in Attention (q)	0.4390	0.5022	0.9142
RPE in Attention (k)	0.4441	0.4892	0.9870
RPE in Attention (v)	0.4890	0.5766	1.0392
RPE in Attention (q, k)	0.4555	0.5083	0.9236
RPE in Attention (q, v)	0.5059	0.5623	0.9546
RPE in Attention (q, k, v)	0.5488	0.6227	1.0977
Ours (k, v)	**0.4102**	**0.4695**	**0.8304**

E. Analysis of Heterogeneous Graph Design

In this section, we examine the following design choice.

1) *Homogeneous Nodes and Edges:* We merge all node and edge types, and let them share the same parameters, in order to examine the necessity of using different parameters for different node and edge types.
2) *Ignoring Vehicle Uncertainty:* We directly weight neighboring vehicles using attention weights, in order to

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RESULTS OF HETEROGENEOUS GRAPH DESIGN

Models	meanATE↓	rmseATE↓	p95 ↓
Homogeneous Nodes and Edges	0.4996	0.6462	1.5224
Ignoring Vehicle Uncertainty	0.4881	0.5295	0.8976
Without Global Type Importance	0.5983	0.6834	1.0801
Ours	**0.4102**	**0.4695**	**0.8304**

TABLE V
RESULTS OF DIFFERENT LOSS DESIGN STRATEGIES

Loss Strategy	rmseATE↓	rmseHorizontal ↓	rmseVertical ↓
MSE	0.4910	0.4741	0.1278
NMSE	1.1792	1.1767	**0.0766**
F-MTL($\gamma = 0.8$)	0.8523	0.8388	0.1507
F-MTL($\gamma = 0.2$)	0.8618	0.8552	0.1064
Ours	**0.4335**	**0.4229**	0.0952

examine the necessity of adjusting fusion weights according to the uncertainty of vehicle node states.

3) *Without Global Type Importance:* We adopt an aggregation method, which is similar to [35] and uses only the importance of different relations to the target node to obtain the final node features, in order to examine the necessity of considering type importance from a global perspective.

The results in Table IV lead to the following conclusions.

1) Not considering global type importance causes the largest performance degradation. A possible reason is that relying only on the importance of local neighbors may trap the model in a local optimum.
2) Using the same parameters for different types of nodes and edges results in the second-largest performance drop. This demonstrates the necessity of using dedicated parameters to model different types.
3) Ignoring the uncertainty of neighboring vehicles causes the smallest performance degradation. Nevertheless, this still confirms the necessity of adjusting the weights according to vehicle node uncertainty so that more reliable cooperative edges can be selected.

F. Analysis of Adaptive Multitask Loss

In Section IV-C, we discuss the importance of the loss function in improving model performance. Here, we compare three different loss design strategies.

1) *MSE:* A unified mse is directly applied to state components of all dimensions.
2) *Normalized mse (NMSE):* The state components of each dimension are first normalized according to their statistical scales, after which the mse is computed.
3) *Fixed Weight Multitask Loss (F-MTL):* The state estimation problem is divided into two subtasks, namely, horizontal position and height, and the losses of the two subtasks are linearly combined using preset fixed weights γ: $L_{\text{total}} = \gamma L_{\text{horizontal}} + (1-\gamma) L_{\text{vertical}}$.

Table V compares different loss design strategies. To further analyze the influence of different loss functions on the optimization of individual state components, this section additionally introduces two metrics, namely, rmseHorizontal and rmseVertical, to evaluate the estimation accuracy in the horizontal plane and along the vertical direction, respectively. This helps further reveal the ability of different loss function designs to balance the optimization of horizontal and vertical position components, while rmseATE is used as the primary evaluation metric.

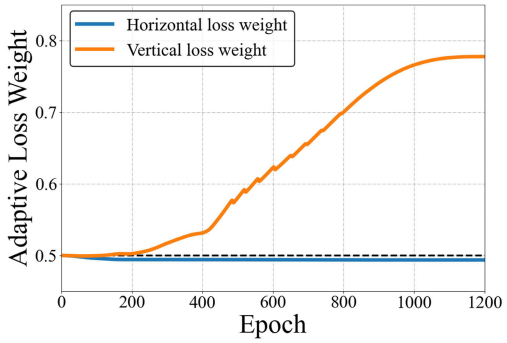


Fig. 6. Evolution of adaptive loss weights during training.

The loss function proposed in this article achieves the best results on both rmseATE and rmseHorizontal, and the second-best performance on rmseVertical. After normalizing the state components in each dimension according to their statistical scales, NMSE places greater emphasis on optimizing the vertical localization component and, therefore, attains the lowest rmseVertical. However, it significantly degrades the estimation performance in the horizontal direction, ultimately leading to the worst overall 3-D localization accuracy in terms of rmseATE. This indicates that properly coordinating the optimization of the horizontal and vertical subtasks is crucial for improving overall localization performance. The F-MTL also performs unsatisfactorily, indicating that manually specified coefficients are insufficient to effectively balance the optimization of the horizontal and vertical subtasks. Although mse provides relatively strong overall performance, it does not account for scale differences across state components. In contrast, the proposed adaptive weighting strategy achieves a more reasonable tradeoff across different components and, therefore, delivers the best overall results.

To explain the superior overall performance of the proposed adaptive multitask loss, Fig. 6 shows the evolution of the two task weights during training. The weights of the horizontal and vertical position losses exhibit clearly different evolution trends during training. The weight of the horizontal loss decreases slightly and remains stable after the early stage of training, whereas the weight of the vertical loss gradually increases and eventually stabilizes. These results indicate that the adaptive loss, while optimizing the horizontal localization error, progressively increases the contribution of the vertical error to the joint optimization process, thereby preventing the smaller-scale vertical loss from being dominated by the horizontal error. As a result, by automatically learning the optimal weight allocation between the two during training, the proposed method achieves a more reasonable optimization balance between horizontal and vertical localization, and, therefore, attains the best overall rmseATE.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

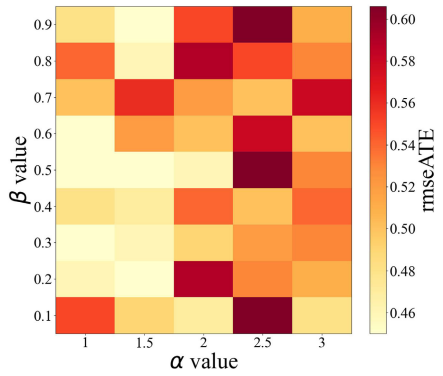


Fig. 7. Results of α and β sensitivity.

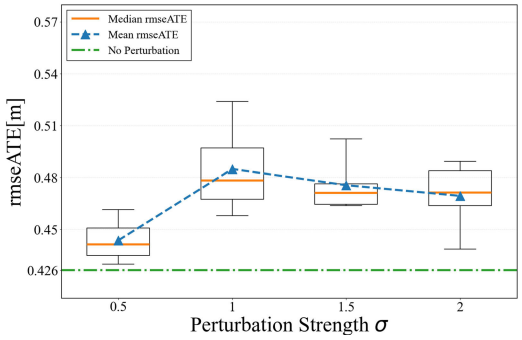


Fig. 8. Performance under different perturbation strengths.

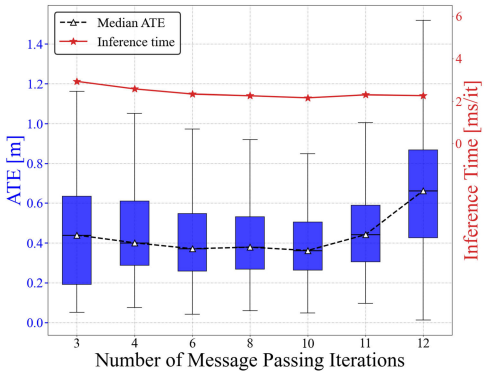


Fig. 9. Performance with different numbers of message passing iterations.

G. Hyperparameter Sensitivity

Fig. 7 presents a 2-D sensitivity analysis of α and β using rmseATE as the evaluation metric. Overall, the model maintains stable performance across a broad range of parameter combinations. This suggests that the proposed method is reasonably robust to the choice of hyperparameters. A closer inspection shows that smaller to moderate α values, around 1.0 to 2.0, together with moderate β values, around 0.3 to 0.6, are typically associated with lower error regions. This indicates that, within this parameter range, the model can suppress noise propagation while preserving more informative cooperative messages. In contrast, when α increases to around 2.5, the rmseATE rises noticeably for most β settings. This observation suggests that an excessively large α may make the fusion behavior overly aggressive. Overall, these results indicate that selecting a small to moderate α together with a moderate β is preferred to obtain higher localization accuracy.

H. Sensitivity Analysis of Uncertainty Inputs

We analyze the robustness of the model to errors in the covariance estimates used to adjust the weights of neighboring vehicles. To simulate inaccurate calibration, we first generate random log-normally distributed perturbations. Specifically, the random perturbation factor is defined as $r_j = \exp(\varepsilon_j)$, where $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$, and σ is the perturbation strength parameter. A larger σ leads to a greater dispersion and a heavier tail of the perturbation factor. After computing the trace of the covariance matrix, the perturbation factor r_j is multiplied by the trace so as to perturb the uncertainty input $u_j^{t-1} = r_j * u_j^{t-1}$. This multiplicative design ensures that the perturbed uncertainty remains positive while also simulating cases in which the covariance is either underestimated or overestimated. We set $\sigma \in \{0.5, 1.0, 1.5, 2.0\}$ and conduct independent experiments with five different random seeds under each perturbation level.

As shown in Fig. 8, when no perturbation is applied, the model achieves an rmseATE of 0.426 m. When the perturbation strength is $\sigma = 0.5$, the model obtains the lowest mean rmseATE. When the perturbation reaches a moderate to high level, for example, when σ is further increased to 1.5 and 2.0, the rmseATE does not show a trend of continuous deterioration as the perturbation strength increases. These results indicate that the accuracy of covariance estimation directly affects the uncertainty-aware vehicle–vehicle aggregation function. At the same time, rmseATE does not exhibit obvious divergence under different perturbation conditions, suggesting that the model is reasonably robust to bias or corruption in the covariance input.

I. Efficiency Analysis

We further investigate the performance of HCLGT. Specifically, we conduct experiments on HCLGT with different numbers of message passing iterations. Each inference estimates only the CL heterogeneous graph at a single time step, that is, the batch size is 1. We also report the runtime of each message passing iteration of the model.

As shown in Fig. 9, appropriately increasing the number of message passing iterations allows the model to more fully aggregate multihop neighborhood information, thereby reducing the error. The performance gradually saturates at around eight to ten message passing iterations. Further increasing the number of iterations instead leads to performance degradation, which may be attributed to oversmoothing of node representations caused by repeated message passing. Meanwhile, the average inference latency of each message passing iteration is approximately 2.39 ms. indicating that the proposed method has good potential for real-time applications. It is worth emphasizing that the current implementation has not been specifically optimized for real-time deployment, and there is still room for further latency reduction.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
RESULTS OF THE GENERALIZATION EVALUATION

Methods	meanATE↓	rmseATE↓	p95 ↓
EKF	2.24±0.06	2.35±0.02	3.18±0.30
VBRDCL	1.63±0.14	1.71±0.09	2.37±0.02
GraphSAGE	1.52±0.02	1.87±0.02	1.87±0.40
Hetero-GNN	1.42± 0.05	1.52±0.02	2.29±0.03
MPNN	1.36±0.05	1.49±0.07	1.94±0.53
Graph Transformer	1.01± 0.08	1.18±0.11	1.99±0.46
GAT	1.03±0.01	1.14±0.04	1.84±0.09
HCLGT(Ours)	**0.36±0.003**	**0.41±0.001**	**0.64±0.001**

J. Generalization Evaluation

To further evaluate whether the performance of HCLGT depends on a single train–validation–test split, we conduct a generalization assessment on the LUCOOP dataset. Since LUCOOP is a continuously collected time series dataset without predefined subsets, threefold sliding cross-validation is adopted to avoid future information leakage: the test set is always located after the training set on the time axis, and the training set and the validation set are sliding-divided on the original training data. Specifically, the first fold uses the first 80% of the original training data as the training set, the second fold uses the first 90%, and the third fold uses the complete training data. The validation set for each fold takes the last 20% of the corresponding training set for early stopping and learning rate scheduling. For learning-based methods, each model is retrained under each fold and evaluated on the corresponding test segment. For nonlearning baselines that do not require training, including EKF and VBRDCL, evaluation is performed directly on the same test segments using the same initialization and measurement inputs. All experimental results report the mean and standard deviation.

Table VI reports the results of the threefold cross-segment generalization evaluation. It can be seen that HCLGT yields small standard deviations on all metrics under different train–validation–test splits, indicating that its performance remains stable across different temporal sliding partitions rather than being caused by chance factors in a single data split. In contrast, some baseline methods, such as Graph Transformer, exhibit much larger variability, with the standard deviation of rmseATE reaching as high as 0.11, indicating poor performance stability.

V. CONCLUSION

This article proposes HCLGT, a Graph Transformer-based CL framework for CAVs in real-world driving environments. HCLGT addresses the relative geometric constraints, heterogeneous cooperative relations, and scale discrepancies among vehicle state components in CL. Specifically, it incorporates edge-derived relative position encoding into dot-product attention, enabling relative measurements to directly guide neighbor matching and information aggregation. It further introduces a two-stage heterogeneous aggregation mechanism to model relation-specific interactions and adaptively fuse information from vehicles and landmarks, together with uncertainty-aware weighting to improve the reliability of neighboring vehicle information. An adaptive multitask loss is also developed to balance the optimization of horizontal and vertical localization components. Extensive experiments on the real-world LUCOOP dataset demonstrate that HCLGT consistently outperforms conventional filtering methods and representative graph learning baselines while exhibiting stable performance across different evaluation settings. These results demonstrate the effectiveness of task-specific heterogeneous Graph Transformer modeling and provide a promising framework for reliable cooperative information fusion in connected autonomous vehicle localization systems.

REFERENCES

[1] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (CAVs)," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6240–6259, Jul. 2022.

[2] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.

[3] D. Li, H. Deng, T. Yu, and L. Zhang, "Multivehicle cooperative localization using a TOA-based simulated annealing extended Kalman filter in urban canyons," *IEEE Internet Things J.*, vol. 12, no. 13, pp. 22832–22846, Jul. 2025.

[4] Y. Lu, H. Ma, E. Smart, and H. Yu, "Real-time performance-focused localization techniques for autonomous vehicle: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6082–6100, Jul. 2022.

[5] Y. Fang et al., "Anomaly diagnosis of connected autonomous vehicles: A survey," *Inf. Fusion*, vol. 105, May 2024, Art. no. 102223.

[6] L. Gao et al., "Cooperative localization in transportation 5.0," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 3, pp. 4259–4264, Mar. 2024.

[7] W. Zhao, S. Zhao, G. Zhang, G. Liu, and W. Meng, "FL-EKF-based cooperative localization method for multi-AUVs," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 30742–30753, Oct. 2024.

[8] J. Xiong, X.-P. Xie, Z. Xiong, Y. Zhuang, Y. Zheng, and C. Wang, "Unified cooperative localization via augmented factor graph and error state message passing," *IEEE Internet Things J.*, vol. 12, no. 19, pp. 40206–40220, Oct. 2025.

[9] J. Xiong et al., "Integrity for belief propagation-based cooperative positioning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11345–11358, Sep. 2024.

[10] J. Xiong, Z. Xiong, Y. Zhuang, J. W. Cheong, and A. G. Dempster, "Fault-tolerant cooperative positioning based on hybrid robust Gaussian belief propagation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6425–6431, Jun. 2023.

[11] M. Liang and F. Meyer, "Neural enhanced belief propagation for cooperative localization," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2021, pp. 326–330.

[12] C. Morris et al., "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4602–4609.

[13] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[14] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," 2020, *arXiv:2006.05205*.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[16] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[17] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," 2020, *arXiv:2012.09699*.

[18] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: A survey of techniques, evaluations and applications," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8003–8042, Aug. 2023.

[19] X. Jia, P. Wu, L. Chen, Y. Liu, H. Li, and J. Yan, "HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13860–13875, Nov. 2023.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.

[20] R. Mendrzik and G. Bauch, "Position-constrained stochastic inference for cooperative indoor localization," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 3, pp. 454–468, Sep. 2019.
[21] W. Shule, C. M. Almansa, J. P. Queralta, Z. Zou, and T. Westerlund, "UWB-based localization for multi-UAV systems and collaborative heterogeneous multi-robot systems," *Proc. Comput. Sci.*, vol. 175, pp. 357–364, Jan. 2020.
[22] A. Wong, Z. Tang, F. J. Jiang, K. H. Johansson, and J. MÃĪrtensson, "Beyond line-of-sight: Cooperative localization using vision and v2x communication," 2025, *arXiv:2507.20772*.
[23] Y. Zou, C. G. Brinton, and V. Tripathi, "Communication-efficient cooperative localization: A graph neural network approach," 2025, *arXiv:2504.08135*.
[24] L. Chen, F. Zheng, X. Gong, and X. Jiang, "GNSS high-precision augmentation for autonomous vehicles: Requirements, solution, and technical challenges," *Remote Sens.*, vol. 15, no. 6, p. 1623, Mar. 2023.
[25] B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, "Message passing neural network versus message passing algorithm for cooperative positioning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 6, pp. 1666–1676, Dec. 2023.
[26] B. C. Tedeschini, M. Brambilla, L. Barbieri, and M. Nicoli, "Addressing data association by message passing over graph neural networks," in *Proc. 25th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2022, pp. 01–07.
[27] H. Jung, S.-W. Ko, and S. Kim, "Toward cooperative localization with implicit connectivity: Graph neural network approach," *IEEE Wireless Commun. Lett.*, vol. 14, no. 10, pp. 3184–3188, Oct. 2025.
[28] L. Lian, C. Bai, Y. Xu, H. Dong, R. Cheng, and S. Zhang, "Learning to beamform for cooperative localization and communication: A link heterogeneous GNN-based approach," *IEEE Trans. Wireless Commun.*, vol. 25, pp. 6177–6190, 2026.
[29] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 6531–6543.
[30] Z. Zhang, A. Liniger, C. Sakaridis, F. Yu, and L. V. Gool, "Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 57481–57499.
[31] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 464–468.
[32] W. Park, W. Chang, D. Lee, J. Kim, and S.-w. Hwang, "GRPE: Relative positional encoding for graph transformer," 2022, *arXiv:2201.12787*.
[33] X. Wang et al., "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, 2019, pp. 2022–2032.
[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 593–607.
[35] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, New York, NY, USA, Apr. 2020, pp. 2704–2710.
[36] Y. Liu et al., "A novel multimodal long-term trajectory prediction scheme for heterogeneous user behavior patterns," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 13275–13291, Dec. 2024.
[37] J. Gao et al., "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11525–11533.
[38] M. Liang et al., "Learning lane graph representations for motion forecasting," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 541–556.
[39] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, Jul. 2022.
[40] Z. Sun, Z. Wang, L. Halilaj, and J. Luettin, "SemanticFormer: Holistic and semantic traffic scene representation for trajectory prediction using knowledge graphs," *IEEE Robot. Autom. Lett.*, vol. 9, no. 9, pp. 7381–7388, Sep. 2024.
[41] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*.
[42] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9120–9132.
[43] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6555–6564.
[44] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.
[45] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "CoordiNet: Uncertainty-aware pose regressor for reliable vehicle localization," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1848–1857.
[46] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022.
[47] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "TrafficBots: Towards world models for autonomous driving simulation and motion prediction," 2023, *arXiv:2303.04116*.
[48] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10033–10041.
[49] S. Han et al., "Uncertainty-aware graph structure learning," in *Proc. ACM Web Conf.*, Apr. 2025, pp. 4863–4874.
[50] H. Min, Y. Li, X. Wu, W. Wang, L. Chen, and X. Zhao, "A measurement scheduling method for multi-vehicle cooperative localization considering state correlation," *Veh. Commun.*, vol. 44, Dec. 2023, Art. no. 100682.
[51] Z. Wang, D. Yu, Q. Li, S. Shen, and S. Yao, "SR-HGN: Semantic{-} and relation-aware heterogeneous graph neural network," *Expert Syst. Appl.*, vol. 224, Aug. 2023, Art. no. 119982.
[52] Y. Zhao et al., "Learning bi-typed multi-relational heterogeneous graph via dual hierarchical attention networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9054–9066, Sep. 2023.
[53] J. Axmann et al., "LUCOOP: Leibniz university cooperative perception and urban navigation dataset," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–8.
[54] J. Yan, F. Zhu, Y. Huang, and Y. Zhang, "Robust decentralized cooperative localization for multirobot system against measurement outliers," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17934–17947, May 2024.
[55] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1024–1034.
[56] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 11960–11970.



Shengsheng Xing received the B.S. degree in information management and information systems from Lanzhou Jiaotong University, Lanzhou, China, in 2023. He is currently pursuing the Ph.D. degree in transportation engineering with Chang'an University, Xi'an, China.

His current research interests include connected and autonomous vehicles, and cooperative localization.



Haigen Min (Senior Member, IEEE) received the Ph.D. degree in traffic engineering from Chang'an University, Xi'an, China, in 2018.

He is currently a Professor with Chang'an University. His research interests mainly include environment perception, decision-making, and motion control based on emerging sensor technologies of connected and autonomous vehicles (AVs) and intelligent transportation systems (ITSs).

Dr. Min has been selected for Shaanxi Province Young Science and Technology Rising Star and Young Top Talent. He received the best paper awards from the COTA International Conference of Transportation Professionals (CICTP). In addition, he serves as an Associate Editor for *International Journal of Transportation Science and Technology*, *China Journal of Highway and Transport*, and *Journal of Traffic and Transportation Engineering*.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.



Xia Wu received the Ph.D. degree in traffic information engineering and control from Chang'an University, Xi'an, China, in 2020.

She is currently an Associate Professor with Chang'an University. Her current research interests include mixed traffic flow control, trajectory optimization with connected and automated vehicles, and human-driven vehicles.



Wuqi Wang (Graduate Student Member, IEEE) received the B.E. degree in software engineering and the Ph.D. degree in traffic information engineering and control from Chang'an University, Xi'an, China, in 2020 and 2026, respectively.

His research interests include simultaneous localization and mapping (SLAM), place recognition, loop closure detection, high-precision localization, and decision-making for connected and automated vehicles (CAVs).



Xiangmo Zhao (Member, IEEE) received the B.S. degree from Chongqing University, Chongqing, China, in 1987, and the M.S. and Ph.D. degrees from Chang'an University, Xi'an, China, in 2002 and 2005, respectively.

He is currently a Professor with the School of Information Engineering, Chang'an University, and the President of Xi'an University of Architecture and Technology, Xi'an. He is the Vice President of the Joint Laboratory for Connected Vehicles, Ministry of Education, and China Mobile, and Shaanxi Road Traffic Intelligent Detection and Equipment Engineering Technology Research Centre, and is also the Leader of the National Key Subjects-Traffic Information Engineering and Control, Chang'an University. His current research interests include connected vehicles, intelligent transportation systems, and computer science.

Dr. Zhao is the Director of the Information Professional Committee, a member of the Advisory Expert Group of the China Transportation Association, the Vice Chairperson of the Institute of Highway Association on Computer Professional Committee, and the Deputy Director of the Institute of Computer in Shaanxi Province.



Guoqiang Mao (Fellow, IEEE) has published over 200 papers in international conferences and journals, which have been cited more than 8500 times. His research interests include intelligent transport systems, applied graph theory and its applications in telecommunications, the Internet of Things, wireless sensor networks, wireless localization techniques, and network modeling and performance analysis.

Dr. Mao is a fellow of IET. He was the Co-Chair of IEEE Intelligent Transport Systems Society Technical Committee on Communication Networks. He has served as the Chair, the Co-Chair, and a TPC member of a number of international conferences. He has been an Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (since 2018), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2014–2019), and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY (since 2010) and received the "Top Editor" award for outstanding contributions to IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2011, 2014, and 2015.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:10:15 UTC from IEEE Xplore. Restrictions apply.