http://dx.doi.org/10.35596/1729-7648-2025-23-4-109-117

UDC 004.896:004.942

DYNAMIC RELATIONAL GRAPH MODELING FOR MULTI-AGENT MOTION TRAJECTORY PREDICTION

YI TANG, DMITRY YU. PERTSAU

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Abstract. Accurate trajectory prediction of multiple agents is a critical task in the fields of autonomous driving, human-computer interaction, and behavior analysis. However, the dynamic and interactive nature of agent behavior poses significant challenges, since it requires the formation of complex spatio-temporal dependencies and dynamically evolving interactions between agents. A novel approach is proposed for modeling dynamic relational graphs, the core component of which is the attention focus block, taking into account the relative positions of graph-based agents. By considering objects in a scene (e.g., vehicles and road elements) as graph nodes and their interactions as edges, the proposed approach effectively captures both local and global dependencies in a scene and makes a prediction about the future trajectory. The presented approach is evaluated using the Argoverse1 trajectory prediction dataset. Experimental results show that this model outperforms existing methods.

Keywords: multi-agent trajectory prediction, graph neural network, attention block.

Conflict of interests. The authors declare no conflict of interests.

For citation. Yi Tang, Pertsau D. Yu. (2025) Dynamic Relational Graph Modeling for Multi-Agent Motion Trajectory Prediction. *Doklady BGUIR*. 23 (4), 109–117. http://dx.doi.org/10.35596/1729-7648-2025-23-4-109-117.

ДИНАМИЧЕСКОЕ ГРАФОВОЕ МОДЕЛИРОВАНИЕ ДЛЯ МНОГОАГЕНТНОГО ПРЕДСКАЗАНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ

И. ТАН, Д. Ю. ПЕРЦЕВ

Белорусский государственный университет информатики и радиоэлектроники (Минск, Республика Беларусь)

Аннотация. Точное прогнозирование траектории движения нескольких агентов является важнейшей задачей в таких областях, как автономное вождение, взаимодействие человека с компьютером и анализ поведения. Однако динамичность и интерактивность поведения агентов создают значительные проблемы, поскольку требуют формирования сложных пространственно-временных зависимостей и динамически развивающегося взаимодействия между агентами. Предлагается новый подход для моделирования динамических реляционных графов, основным компонентом которых является блок акцента внимания с учетом относительного положения агентов на основе графов. Рассматривая объекты в сцене (например, транспортные средства и элементы дороги) как узлы графа, а их взаимодействие как ребра, предложенный подход эффективно отражает как локальные, так и глобальные зависимости на сцене и делает прогноз о будущей траектории. Представленный подход оценивается с помощью набора данных для прогнозирования траектории Argoverse1. Экспериментальные результаты показывают, что такая модель превосходит существующие методы.

Ключевые слова: многоагентное прогнозирование траектории, графовая нейронная сеть, механизм внимания.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Тан, И. Динамическое графовое моделирование для многоагентного предсказания траектории движения / И. Тан, Д. Ю. Перцев // Доклады БГУИР. 2025. Т. 23, № 4. С. 109–117. http://dx.doi.org/10.35596/1729-7648-2025-23-4-109-117.

Introduction

The rapid advancement of autonomous driving technology is revolutionizing transportation systems, with trajectory prediction serving as a critical component to enhance both vehicle safety and driving efficiency. The goal of trajectory prediction is to accurately forecast the future behavior of traffic participants (e. g., vehicles, pedestrians, cyclists) [1]. This task is highly challenging due to the complex factors involved, including interactions between agents, motion dynamics, and constraints imposed by the map environment.

In the early stages of trajectory prediction research, traditional methods were typically based on vehicle dynamics models, utilizing historical states of agents (e. g., position, velocity, acceleration) to predict future motion trends. Methods such as Kalman Filters, Dynamic Bayesian Networks, and Hidden Markov Model demonstrated satisfactory performance in simple traffic scenario [2]. However, in more complex and dynamic traffic environments with multi-agent interactions, these traditional methods exhibited significant limitations in modeling intricate interaction relationships and capturing long-term dependencies.

With the emergence of deep learning, the field of trajectory prediction has undergone a paradigm shift. The integration of high-definition (HD) maps and sensor data introduced new perspectives for research. By combining map information with sensor data, researchers achieved significant improvements in prediction accuracy [3]. However, this also introduced challenges related to computational complexity and data fusion. Efficiently leveraging such heterogeneous data has become a core research question. Early studies often employed rasterized representations to convert HD maps into grid-like 2D images, enabling convolutional neural networks (CNNs) to extract spatial features. For instance, Casas et al. [4] utilized CNNs to extract road semantic features from rasterized maps, while Hong et al. [5] combined high-resolution 3D perception data with semantic maps to encode spatial characteristics. Although these rasterization-based methods effectively incorporated map data, their large perception range resulted in high computational costs and potential loss of critical map structural information, such as road topology and traffic constraints. To address these limitations, research has gradually shifted toward vectorized representations. These methods encode maps, agents, and other scene elements as vectorized features and leverage permutation-invariant operators, such as point cloud convolutions, graph convolutions, and transformers, to capture scene context. For example, VectorNet [6] as a pioneering work, modeled road maps and agent trajectories in a vectorized manner and utilized graph neural networks (GNNs) to capture interactions between agents, road environments, and other traffic participants. This approach improved the compactness and information retention of map representations, avoiding the information loss associated with rasterization. Building on this, LaneGCN [7] constructed a graph model based on road network topology, representing map elements as nodes and using GNNs to encode multi-level information, thereby explicitly modeling the local connectivity and global interaction relationships of road structures. HiVT [8] proposed a Hierarchical Vectorized Transformer to model multi-granularity interactions between agents.

Despite these advancements, most existing methods overly emphasize interactions between agents and map elements, while neglecting the potential relationships within agents or within map elements. These internal relationships, such as complex multi-agent interactions and internal structural associations of map elements, are crucial for comprehensively modeling the dynamic characteristics of traffic environments. Additionally, existing methods often struggle to effectively integrate local and global dependencies, leading to insufficient context capture.

To address these challenges, we propose a novel framework for Dynamic Relational Graph (DRG) modeling. This framework represents entities in the scene (e. g., vehicles and road elements) as nodes in a graph, with edges capturing their interactions, thereby effectively modeling both local and global dependencies within the scene. By incorporating relative positional information into a multi-head graph attention mechanism, the model efficiently captures spatio-temporal relationships in non-Euclidean feature spaces. This approach enables a more nuanced understanding of scene context and agent interactions, significantly improving trajectory prediction accuracy.

Problem statement

The trajectory prediction task aims to generate potential future trajectories for target agents based on their observed motion history and surrounding map information. Specifically, in a driving scenario with N_a moving agents, we use M to represent the map information and $X = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N_a}\}$ to denote the observed historical trajectories of all agents. For each agent i, its historical trajectory over the past H time steps are represented as $\mathbf{x}_i = \{\mathbf{x}_i^{-H+1}, \mathbf{x}_i^{-H+2}, ..., \mathbf{x}_i^{0}\}$.

The multi-agent motion predictor generates potential future trajectories in future T time steps for all agents in the scene, denoted as $Y = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{N_a}\}$. For each agent i, K possible future trajectories and their corresponding probability scores are predicted to capture the multimodal nature of motion. The predicted trajectories for agent i are represented as $\mathbf{y}_i = \{\mathbf{y}_i^1, \mathbf{y}_i^2, ..., \mathbf{y}_i^K\}$, where each trajectory $\mathbf{y}_i^k = \{\mathbf{y}_{i,1}^k, \mathbf{y}_{i,2}^k, ..., \mathbf{y}_{i,T}^k\}$ ($k \in \{1, 2, ..., K\}$) represents the k^{th} predicted trajectory of the i^{th} agent over future T time steps. The associated probability scores for these trajectories are represented as $\mathbf{s}_i = \{s_i^1, s_i^2, ..., s_i^K\}$.

Method overview

This study proposes a trajectory prediction model based on relative positional feature fusion, leveraging a hierarchical network structure to learn and integrate various semantic relational features. As illustrated in Fig. 1, *a*, the proposed model comprises three primary components: intra-relational feature extraction, lane-actor fusion network, and multi-modal decoder.

We adopt a vectorized scene representation approach. As depicted in Fig. 1, *b*, the process includes scene vectorization and the computation of relative positions. Specifically, for each instance, such as trajectories and lane segments, a local reference frame is constructed to decouple the inherent features of the instances from their relative information. Following [9], a 5-dimensional relative positional vector $r_{i\to j} = [\sin(\alpha_{i\to j}), \cos(\alpha_{i\to j}), \sin(\beta_{i\to j}), \cos(\beta_{i\to j}), \|d_{i\to j}\|]$ between elements i and j is used to represent the spatial relationships between actors and actors, lanes and lanes, as well as actors and lanes in the scene, denoted as $r_{a\to a} \in \mathbb{R}^{N_A\times 5}$, $r_{l\to l} \in \mathbb{R}^{N_L\times 5}$ and $r_{a\to l} \in \mathbb{R}^{N\times 5}$, respectively. Here, $N = N_A + N_L$, N_A indicates the number of actors in the scene, and N_L represents the number of lane elements.

In the next step, the corresponding relative positional information is passed into their respective encoders to fuse and extract fundamental features x_A and x_L. Subsequently, these features, along with the relative positional information in the scene $r_{a\to l}$, are input into the feature fusion stage for comprehensive feature integration. Finally, the fused features are fed into the multi-modal decoder to forecast the trajectories of all target agents.

1. Intra-relational feature extraction

Following LaneGCN [7], our trajectory feature extraction module first employs a one-dimensional CNNs based Feature Pyramid Network (FPN) to encode the historical trajectories of vehicles within the scene denoted as X_A, thereby extracting fundamental features. The 1D CNNs effectively captures local patterns in the temporal dimension of the trajectories, such as acceleration, deceleration, or turning behaviors. Meanwhile, the FPN structure helps to address the diversity of trajectory lengths by extracting multi-scale features, enhancing the model's ability to represent both short and long trajectories. After initial encoding, to further capture the spatiotemporal dependencies of historical trajectories, we utilize a 4-layer Gated Recurrent Unit (GRU) network to process the encoded trajectory features and extract refined representations X_A^r. Compared to Long Short-Term Memory model, GRU offers a simpler structure and higher computational efficiency, making it particularly well-suited for trajectory modeling in complex scenarios. For extracting road features from map tensors, we leverage PointNet [10] to encode the lane nodes and structural information of the road network X_L, thereby obtaining fundamental lane features X_L^r. PointNet directly processes unstructured data points within lanes and effectively models both the local geometric features and the global topological structure of the lanes, which ensures the lane features incorporate critical configuration information, such as lane curvature, branching points, and connections between nodes. This process can be demonstrated in the following equations:

$$X_A^r = GRU^{(4)}(FPN_{1\times 1}(X_A)); \tag{1}$$

$$X_L^r = PN(X_L), \tag{2}$$

where $FPN_{1\times 1}$ refers to 1D CNNs based Feature Pyramid Network; $GRU^{(4)}$ indicates the four-layer GRU operation; PN corresponds to the PointNet network.

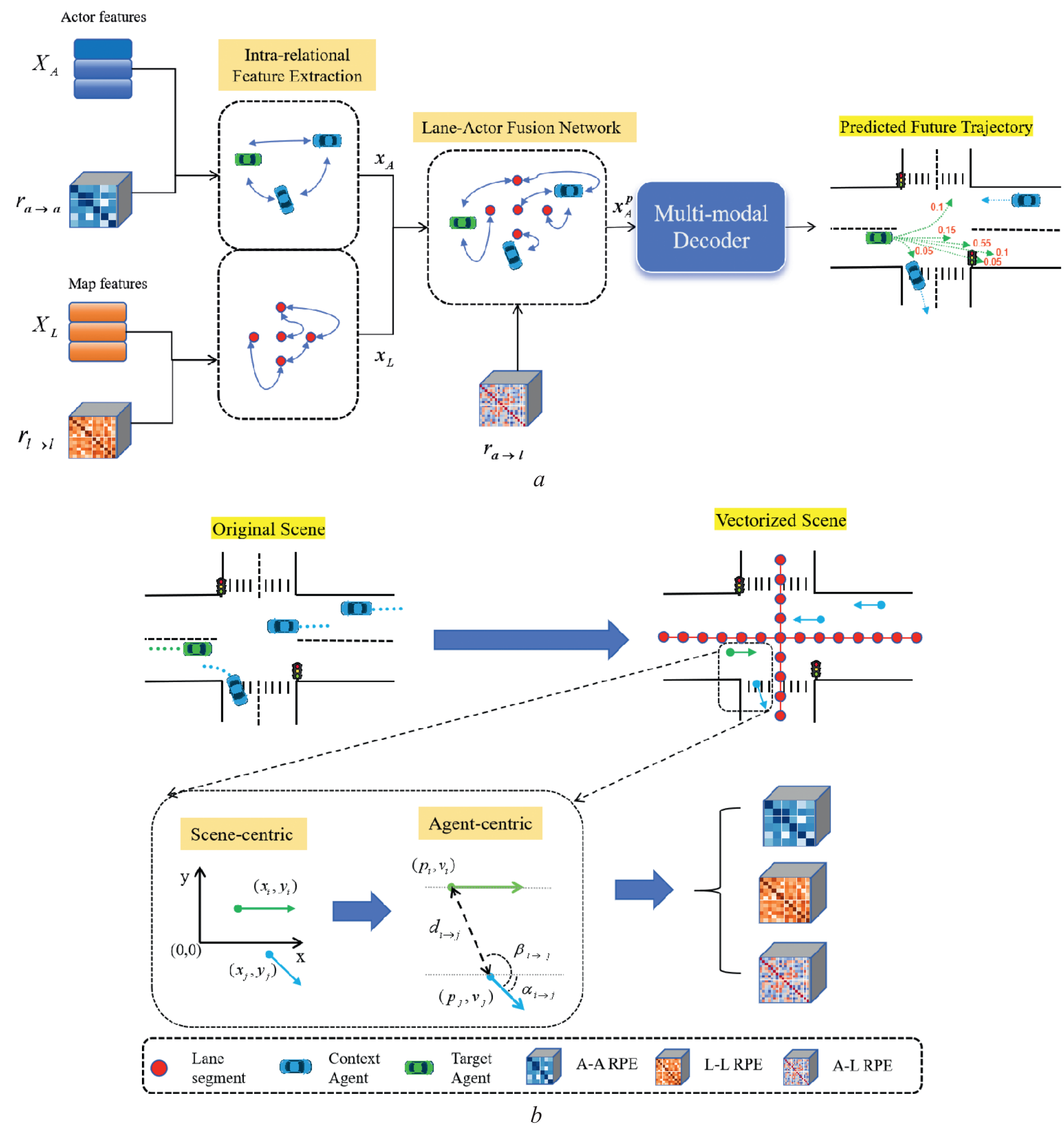

Fig. 1. Illustration of proposed DRG: *a* – architecture and flowchart of proposed model; *b* – scene vectorization and relative position calculation.

Although the aforementioned methods generate relatively rich features, existing approaches often directly utilize these features without explicitly modeling the latent relationships within actors (vehicles) or lanes. These internal relationships are crucial for accurately capturing multi-agent behavior interactions in complex scenarios. To address this limitation, we explicitly introduce relative positional information, which is paired with the corresponding actor or lane features and then fed into the Relative Position-aware Graph Attention Transformer (RP-GAT). Relative position explicitly represents the spatial positioning and semantic relationships between actors or between lanes, such as the relative distance and angles between vehicles, or the topological structure between lane nodes. Through the attention mechanism of RP-GAT, these features are further fused, resulting in more expressive actor and lane features, denoted as x_A and x_L, respectively:

$$\begin{cases} x_A = \xi^{(1)}(X_A^r, r_{a\to a}); \\ x_L = \xi^{(1)}(X_L^r, r_{l\to l}), \end{cases} \tag{3}$$

where $\xi^{(1)}$ is the one-time loop of RP-GAT module.

2. Feature fusion

At first, two separate linear layers are applied to further process the features of actors and lanes obtained from corresponding encoders. These processed features are then concatenated along the first dimension. Subsequently, the concatenated features, combined with the relative positional information of all elements in the scene, are fed into the RP-GAT to produce actor trajectory features that incorporate road element information. The specific details of RP-GAT are illustrated below in Fig. 2. The input features are first passed through three linear layers to obtain Q (Query matrix), K (Key matrix), and V (Value matrix), respectively. Multi-head attention is then applied to compute attention scores between these representations. To incorporate the corresponding relative positional information, the relative positional features are mapped to the same dimensional space through a linear layer and multiplied with the attention scores. This results in scores that comprehensively account for both feature interactions and positional relationships. Finally, a feedforward network further processes these scores through L times (here, L is 4). After each feedforward operation, the Add & Norm module is applied, where the input of the feedforward network is added back to its output, followed by layer normalization. This ensures better gradient flow during training and stabilizes the learning process. Thus obtain the final trajectory features for the actors denoted as x_A^p. The processes can be represented by the equation

$$x_A^p = \xi^{(4)}(MLP^{(2)}(x_A) \oplus MLP^{(2)}(x_L)), \qquad (4)$$

where $MLP^{(2)}$ is the two sequential linear layers; $\oplus$ is the concatenate operation; $\xi^{(4)}$ is the four times loop of RP-GAT module.

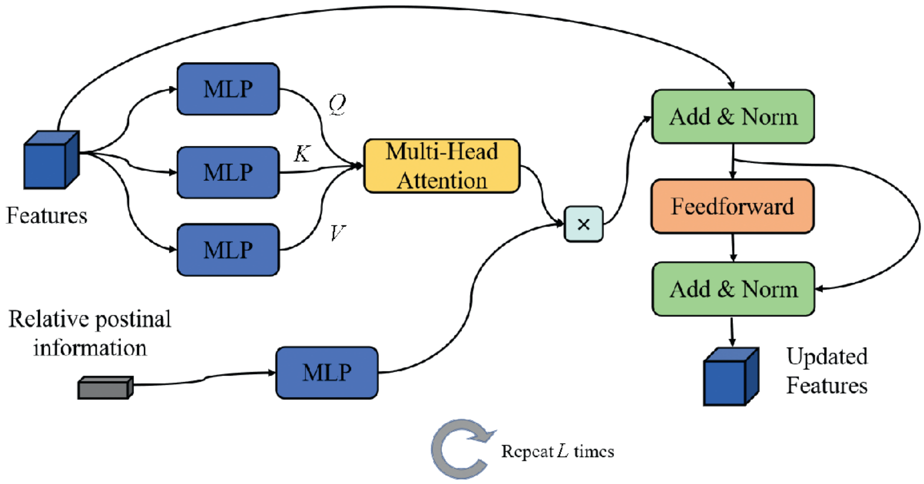

Fig. 2. Architecture of Relative Position-aware Graph Attention Transformer: MLP is the Linear layer with Layer Normalization and ReLU activation function; “×” is the matrix multiplication

3. Multi-modal decoder

During the trajectory generation process, we first utilize the fused actor trajectory features x_A^p to model multi-modal future trajectories at first. To generate K (in this case, $K = 6$) possible future trajectories, the model employs a multi-head attention mechanism to map the embedded features into multiple latent spaces, each corresponding to a distinct trajectory mode. The resulting embedding is denoted as E. This multi-head projection is implemented using two fully connected layers with non-linear activation functions. Then, the embedded vectors E are used to generate confidence scores $\mathbf{S}$ of each mode of each agent as well as predict the final K trajectories denoted as $\mathbf{Y}$. Specifically, three sequential linear layers to generate the trajectories of each agent. To evaluate the confidence of each trajectory mode, the model applies three linear layers and a SoftMax function to generate confidence scores for K trajectories of each agent denoted as s. The processes are shown in equations:

$$\begin{cases} E = MLP^{(2)}(x_A^p); \\ \mathbf{Y} = MLP^{(3)}(E); \\ \mathbf{S} = \chi(MLP^{(3)}(E)), \end{cases} \qquad (5)$$

where $MLP^{(3)}$ is the three sequential linear layers; χ is the SoftMax function.

When predicting future trajectories, directly regressing trajectory points **Y** often leads to physical inconsistencies, such as discontinuities between consecutive points, which deviate from the smoothness observed in real-world trajectories. To address these issues, we adopt a parameterization approach based on monomial basis functions for trajectory generation [9]. Specifically, the trajectory is represented as a polynomial expansion with respect to time t, expressed as $p(t)=\sum_{i=0}^{n} w_i t^i$, where $w_i \in \mathbb{R}^2$ are the polynomial coefficients that control the trajectory's shape, and n is the polynomial order. The n is configurated as 5, consistent with the 5-second time horizon according to [11]. Here t is a normalized time variable that takes values between 0 and 1, corresponding to the evenly spaced time steps over the prediction horizon. The normalization ensures time remains relative and independent of scale. This polynomial-based approach leverages the higher-order differentiability of polynomials to inherently ensure the continuity and smoothness of the generated trajectories, effectively mitigating the physical inconsistencies often encountered in non-parametric prediction methods. Furthermore, this multimodal trajectory generation mechanism dynamically captures various plausible outcomes in complex traffic scenarios while assigning confidence levels to each predicted mode. This not only enhances the interpretability of the predictions but also improves their practical applicability in real-world settings. The detailed implementation can be found at https://github.com/tyjcbzd/DRG.

Experiments and results

Dataset description. We train and test proposed model on Argoverse v1 Motion Prediction Dataset [11], it is a widely used benchmark designed to facilitate research in self-driving motion prediction. It provides high-quality trajectory data for agent vehicles along with semantically rich high-definition map information, enabling comprehensive spatiotemporal modeling. The dataset consists of 324.557 real-world driving scenarios collected from over 1.000 hours of driving in Pittsburgh and Miami. These scenarios are divided into training, validation, and test sets, containing 205.942, 39.472 and 78.143 samples, respectively. Each scenario spans five seconds, sampled at 10 Hz, with the first two seconds provided as historical trajectories of agent vehicles, requiring forecasting models to predict their movements for the subsequent three seconds [9]. The dataset includes both trajectory data and high-definition map elements such as lane centerlines with connectivity information, offering a realistic and challenging benchmark for trajectory prediction in complex urban driving environments.

Implementation details. We adopt a PolylineLR [7] scheduler to control the learning rate dynamically over 40 training epochs. The initial learning rate is set to 1.0e-4, and the values for learning rate adjustments are defined as [1.0e-4, 1.0e-3, 1.0e-3, 1.0e-4], corresponding to the milestones [0, 5, 35, 40]. At the beginning (epochs 0–5), the learning rate increases linearly from 1.0e-4 to 1.0e-3. Epochs 5–35, the learning rate remains constant at 1.0e-3 to ensure stable optimization in the main training phase. Finally, during epochs 35–40, the learning rate decays linearly back to 1.0e-4 to promote convergence. This dynamic scheduling strategy provides a warm-up period, a stable training phase, and a gradual learning rate reduction for smoother convergence. The model is trained on 2 RTX 3090 with a global batch size of 64. In addition, we consider only agents and lane segments within a 50-meter radius of focal agent.

Evaluation metrics. We have adopted the standard testing and evaluation methodology used in motion prediction competitions to assess prediction performance [7]. Key metrics for individual agents include Minimum Final Displacement Error (minFDE), Minimum Average Displacement Error (minADE) and Miss Rate (MR). Here MR measures the percentage of trajectories, where the distance between the predicted final position and the ground truth final position exceeds a predefined threshold d (here d is configurated as 2). minFDE reflect the accuracy of the predicted endpoints, and minADE indicates the overall bias in the predicted trajectories. Calculations were carried out using the following formulas:

$$\min \mathrm{ADE}=\frac{1}{T}\min_{i\in\{1,2,\ldots,K\}}\sum_{t=1}^{T}\sqrt{(\hat{x}_t^i-x_t)^2+(\hat{y}_t^i-y_t)^2}; \tag{6}$$

$$\min \mathrm{FDE}=\min_{i\in\{1,2,\ldots,K\}}\sqrt{(\hat{x}_T^i-x_T)^2+(\hat{y}_T^i-y_T)^2}; \tag{7}$$

$$\mathrm{MR} = \frac{1}{N}\sum_{n=1}^{N} 1\left(\min_{i\in\{1,2,\ldots,K\}} \sqrt{(\hat{x}_T^i - x_T)^2 + (\hat{y}_T^i - y_T)^2} > d\right), \tag{8}$$

where $t \in [0, T]$, $\hat{x}_t, \hat{y}_t$ are the predicted points at time t; $\hat{x}_T^i, \hat{y}_T^i$ are the predicted points of a positive trajectory (e. g. with minimal final displacement error) at time t; N is the total number of all trajectory samples; $1(\cdot)$ is a function that returns 1 if the condition in brackets is true and 0 otherwise.

Quantitative results. Tabl. 1 shows the comparisons with the state-of-the-art methods listed on the leaderboard of the Argoverse 1 motion Forecasting test set (the numbers highlighted in bold represent the best-performing results). Specifically, we compared several trajectory prediction models, including LaneGCN, DenseTNT, THOMAS, HiVT-128, GANet, SceneTrans, MacFormer, and proposed DRG model, evaluating their performance in terms of minFDE, minADE, MR, and model parameters (M). The evaluation focused on both single- and multi-modal trajectory prediction (K is 1 and 6 respectively).

Table 1. Comparisons with the state-of-the-art methods listed on the leaderboard of the Argoverse 1 motion Forecasting test set

Methods	minFDE		minADE		MR	Param/M
	$K = 6$	$K = 1$	$K = 6$	$K = 1$	$K = 6$	
LaneGCN [7]	1.36	3.76	0.87	1.70	0.162	3.7
DenseTNT [12]	1.38	3.69	0.91	1.70	0.125	–
THOMAS [13]	1.44	3.69	0.94	1.67	**0.104**	–
HiVT-128 [8]	1.17	3.53	0.77	1.60	0.127	2.5
GANet [14]	1.16	3.46	0.81	1.59	0.118	2.4
SceneTrans [15]	1.24	4.57	0.80	1.75	0.126	15.3
MacFormer [16]	**1.22**	3.72	0.82	1.70	0.120	2.4
DRG (ours)	1.24	**2.89**	**0.73**	**1.46**	0.109	2.2

Experimental results demonstrate that our model achieves superior performance across multiple metrics, especially in multi-modal prediction ($K = 6$). Our model achieves a minADE of 0.73, outperforming all comparison models, including HiVT-128 (0.77) and GANet (0.81). In single-modal prediction ($K = 1$), our model achieves a minADE of 2.89, significantly better than other methods, such as GANet (3.46) and DenseTNT (3.69). These results indicate that our model not only exhibits higher accuracy in single trajectory prediction, but also excels in scenarios requiring the modeling of diverse potential trajectory distributions. By more accurately capturing the actual movement trends of the target, our model substantially reduces prediction errors in multi-modal scenarios. Furthermore, our model demonstrates strong performance in terms of miss rate (MR). In the multi-modal prediction task, the MR reduces to 0.109, one of the lowest values among all compared models. This significant reduction in miss rate highlights the model's higher confidence and robustness in predicting target trajectories, effectively reducing the instances of failed predictions. This improvement is particularly crucial in real-world applications, where safety is paramount, such as autonomous driving and robotic navigation, as it significantly enhances system reliability and stability. In addition to its prediction accuracy, our model exhibits an efficient lightweight design. With a parameter size of just 2.2M, it is significantly smaller than other high-performance models, such as SceneTrans (15.3M) and LaneGCN (3.7M). This compact design not only reduces storage and computational costs, but also improves operational speed and deployment efficiency in real-world applications.

Qualitative results. Trajectory predictions in four scenarios of varying complexity are illustrated in Fig. 3. The leftmost image and the bottom-middle image showcase the performance of our model in long-distance scenarios, where it provides accurate and precise trajectory predictions even over extended horizons. This robustness demonstrates the model's capability to maintain reliable predictions in situations, where long-term positional trends are critical. In the top-middle image, the scene features a highly complex and interactive intersection involving multiple agents. Our model successfully predicts trajectories for these agents under different intents, such as going straight or making turns, while maintaining robustness across all possible actions. The accurate handling of complex interactions between agents highlights the model's ability to navigate densely interactive environments with high prediction confidence and minimal error. The rightmost image depicts a three-pronged intersection, where the intent of the vehicles is less clear. In this scenario, our model produces diverse and multimodal predictions, spreading the predicted trajectories across various potential outcomes. These predictions include not only straight-ahead motions, but also turning behaviors, all conditioned on the actual road topology and environment constraints. This capability of capturing trajectory uncertainty in ambiguous scenarios ensures that the model remains flexible and adaptive to diverse possibilities, further enhancing its practical utility.

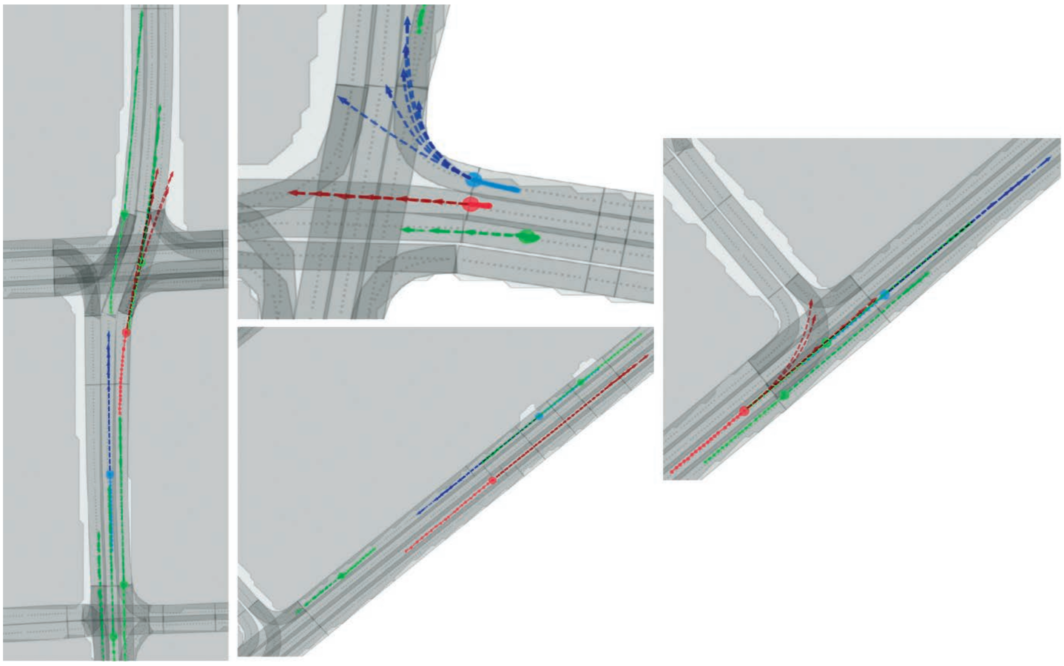

Fig. 3. Qualitative results on the test set of Argoverse1

Conclusion

1. A novel framework, Dynamic Relational Graph Modeling, for multi-agent trajectory prediction is proposed. The core component is the Relative Position-aware Graph Attention Transformer, which dynamically integrates graph attention mechanisms and spatiotemporal position encoding to model inter-agent dependencies with precision. Leveraging the modular nature of the transformer architecture, our framework is capable of capturing long-range interactions and enhancing multi-layer trajectory representations.

2. Experimental results on the Argoverse1 dataset show superior performance and interpretability in analyzing evolving inter-agent relationships. The ability to explicitly model dynamic relational structures positions our approach as an ideal tool for dynamic systems in real-world scenarios requiring fine-grained trajectory prediction, such as autonomous driving, human-computer interaction, and collaborative robotics. Future work will focus on extending hierarchical modeling, improving computational efficiency for real-time deployment, and validating robustness across diverse datasets.

3. The source code of the project has been published and is available for free non-commercial use: https://github.com/tyjcbzd/DRG.

References

1. Yurtsever E., Lambert J., Carballo A., Takeda K. (2020) A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*. 8, 58443–58469.
2. Wang S., Bao Z., Culpepper J. S., Cong G. (2021) A Survey on Trajectory Data Management, Analytics, and Learning. *ACM Computing Surveys (CSUR)*. 54 (2), 1–36.
3. Singh A. (2023) Trajectory-Prediction with Vision: A Survey. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3318–3323.
4. Casas S., Luo W., Urtasun R. (2018) IntentNet: Learning to Predict Intention from Raw Sensor Data. *In Proceedings of the Conference on Robot Learning. Zürich, Switzerland*. 947–956.

5. Hong J., Sapp B., Philbin J. (2019) Rules of the Road: Predicting Driving Behavior with a Convolutional Model of Semantic Interactions. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA*. 8454–8462.
6. Gao J., Sun C., Zhao H., Shen Y., Anguelov D., Li C., et al. (2020) VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 11525–11533.
7. Liang M., Yang B., Hu R., Chen Y., Liao R., Feng S., et al. (2020) Learning Lane Graph Representations for Motion Forecasting. *In Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, Aug. 23–28, 2020, Proceedings, Part II. Springer International Publishing.* 541–556.
8. Zhou Z., Ye L., Wang J., Wu K., Lu K. (2022) HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8823–8833.
9. Zhang L., Li P., Liu S., Shen S. (2024) SIMPL: A Simple and Efficient Multi-Agent Motion Prediction Baseline for Autonomous Driving. *IEEE Robotics and Automation Letters.*
10. Qi C. R., Su H., Mo K., Guibas L. J. (2017) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 652–660.
11. Chang M.-F., Lambert J., Sangkloy P., Singh J., Bak S., Hartnett A., et al. (2019) Argoverse: 3D Tracking and Forecasting with Rich Maps. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8748–8757.
12. Gu J., Sun C., Zhao H. (2021) DenseTNT: End-to-End Trajectory Prediction from Dense Goal Sets. *In Proceedings of the IEEE/CVF International Conference on Computer Vision.* 15303–15312.
13. Gilles T., Sabatini S., Tsishkou D., Stanciulescu B., Moutarde F. (2022) THOMAS: Trajectory Heatmap Output with Learned Multi-Agent Sampling. *In International Conference on Learning Representations.*
14. Wang M., Zhu X., Yu C., Li W., Ma Y., Jin R., et al. (2023) GANet: Goal Area Network for Motion Forecasting. *In 2023 IEEE International Conference on Robotics and Automation (ICRA).* 1609–1615.
15. Ngiam J., Vasudevan V., Caine B., Zhang Z., Chiang H. T. L., Ling J., et al. (2023) Scene Transformer: A Unified Architecture for Predicting Future Trajectories of Multiple Agents. *In International Conference on Learning Representations.*
16. Feng C., Zhou H., Lin H., Zhang Z., Xu Z., Zhang C., et al. (2023) MacFormer: Map-Agent Coupled Transformer for Real-Time and Robust Trajectory Prediction. *IEEE Robotics and Automation Letters.*

Received: 18 March 2025

Accepted: 22 July 2025

Author's contribution

Yi Tang conducted all aspects of the study independently, including task setting, sample preparation, plotting, and manuscript writing.

Pertsau D. Yu. provided essential assistance by thoroughly reviewing the manuscript, offering constructive feedback, and ensuring the accuracy and clarity of the content.

Information about the authors

Yi Tang, Postgraduate at the Department of Information Technologies in Automated Systems, Belarusian State University of Informatics and Radioelectronics

Pertsau D. Yu., Cand. Sci. (Tech.), Associate Professor, Associate Professor at the Department of Electronic Computer Science, Belarusian State University of Informatics and Radioelectronics

Address for correspondence

220013, Republic of Belarus,
Minsk, Brovki St., 6
Belarusian State University
of Informatics and Radioelectronics
Tel.: +375 25 764-41-91
E-mail: tangyijcb@163.com
Yi Tang