Density-Adaptive Model Based on Motif Matrix for Multi-Agent Trajectory Prediction

Supplementary Material

7. CVAE Structure

To enhance the accuracy and generalization ability of our proposed blocks, we employ the CVAE framework for optimization.

Prior & Encoder. The encoding of local agent trajectories and global trajectory information is crucial for our proposed blocks, and we achieve this by first using an MLP to embed the original trajectory and then utilizing an LSTM to extract the original trajectory feature of each agent in Eq (9). Notably, the MLP and LSTM used for extracting φ_i are shared among all agents. We then extract the global trajectory feature φ_G using the PointNet backbone structure followed by an MLP, as shown in Eq. (10)

$$\varphi_i = \text{LSTM}\left[\text{MLP}\left(X_i\right)\right], \tag{9}$$

$$\varphi_G = \text{MLP}\left[\text{PointNet}\left(X_1; \cdots; X_N\right)\right]. \tag{10}$$

The Gaussian parameters $\mu_\theta, \sigma_\theta$ can be derived through the prior network in Eq.(1), with feature os lane φ_L and φ_{GAT} serving as input feature vectors, that is

$$\mu_\theta, \sigma_\theta = \text{MLP}_\theta\left(\varphi_i; \varphi_L; \varphi_{GAT}\right).$$

Note that φ_L is extracted using the same structure as Eq.(9).

Posterior & Decoder. The decoder network uses an LSTM network to iteratively predict the future position of each time step, as shown below:

$$\begin{aligned}
\rho_i^t &= \text{MLP}_{\text{emb}}\left(\varphi_i, \varphi_L, \varphi_{GAT}, \Phi_i, \overline{p_i^t}\right), \\
h_i^{t+1} &= \text{LSTM}\left(\rho_i^t, \varphi_L; \varphi_{GAT}; \Phi_i, h_i^t\right), \\
\overline{p_i^{t+1}} &= \text{MLP}_{\text{dec}}\left(h_i^{t+1}\right).
\end{aligned}$$

Here, $\overline{p_i^t}$ represents the predicted future trajectory of each agent; ρ_*^t and h_*^t are the process vector for embedding and hidden states. For the approximated posterior p_τ in Eq.(**??**), we use the same structure as the prior network. However, the ground truth of the future trajectory Y_i is passed in the posterior network during the training phase. On the other hand, Y_i is not available in the inference phase. Specifically:

$$\mu_p, \sigma_p = \text{MLP}_p\left(Y_i; \varphi_i; \varphi_L; \varphi_{GAT}\right),$$

Evidence of Lower bound We train our model by optimizing the negative evidence lower bound (ELBO) objective, which involves maximizing the log-likelihood of the training data while minimizing the Kullback-Leibler (KL) divergence between the approximate posterior distribution p_τ in Eq. (1) and the prior distribution p_θ. Specifically, we maximize the objective function:

$$\begin{aligned}
L_{\text{ELBO}} = &-\mathbb{E}_\tau\left\{\log\left[p\left(Y|\Phi, X, Z\right)\right]\right\} \\
&+ \text{KL}\left[p\left(\Phi|Y, X, Z\right) \| p\left(\Phi|X, Z\right)\right].
\end{aligned}$$

Here $p\left(\Phi|Y, X, Z\right)$ is p_τ as mentioned, we use a neural network to approximate the posterior distribution, where the latent codes Φ are inferred from the future trajectories Y of all agents. Similarly, Y can be modeled by Φ using p_D.

8. Graph Attention Network

We extract the node feature Φ_{s_i} and the edge feature Φ_t from ASI and ATI, respectively. These features are then used to construct a bidirectional fully-connected GNN that attentively models the interaction between the ego vehicle and its neighboring vehicles. We utilize GAT [47] as our graph convolutional operator to attentively fuse multi-agent interactions as follows

$$v_i^{(m+1)} = \lambda_{i,i}^{(m)} \cdot v_i^{(m)} + \sum_{j \in \mathcal{N}_i} \lambda_{i,j}^{(m)} \cdot v_j^{(m)}, \tag{11}$$

where v_* is the node feature of agent i at m-th layer of GNN. Unlike other graph convolution operators, GAT is specifically designed to fuse multi-agent interactions attentively by balancing the tradeoff between local and global interactions.Therefore, the initial node feature $v_i^{(0)}$ consists of three parts as follows

$$v_i^{(0)} = \left(X_i\ ; \varphi_i\ ; \Phi_{s_i}\right).$$

In Eq.(11), the set of neighbors when agent i is given is denoted as $\mathcal{N}_i = \mathcal{N} \backslash \{i\}$. The final feature $v_i^{(f)}$ of each node can be expressed as

$$v_i^{(f)} = \text{LN}\left\{\text{FC}\left[v_i^{(\mathcal{G})}\right]\right\},$$

where $\mathcal{G}$ is the fine layer of each agent i. The feature of edges e_* is computed in λ, where both $\lambda_{i,i}^{(m)}$ and $\lambda_{i,j}^{(m)}$ in Eq.(11) are attention coefficients, there is

$$\lambda_{i,j}^{(m)} = \frac{\exp\left\{\text{LeakyReLU}\ \left[a'\left(Wv_i; Wv_j; We_{i,j}\right)\right]\right\}}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \exp\left\{\text{LeakyReLU}\ \left[a'\left(Wv_i; Wv_k; We_{i,k}\right)\right]\right\}}.$$

the $k \in \mathcal{N}_i \cup \{i\}$ in the above equation represents the neighbor information of the node, followed by i to calculate the attention score and consider its own information. We use a' as a learnable parameter for an inner product mapping. The feature of an edge from node i to j is denoted as

$$e_{i,j} = (\sum_{t=t_1}^{T} p_j^t - p_i^t; \Phi_{t_i}^{i,j}),$$

where i, j in superscript is index of row and column. The dimension of the attention coefficients is satisfied $\dim(a') = 2\dim(v_i) + \dim(e_{i,j})$. Lastly, we obtain the final feature of GAT, denoted as φ_{GAT}, by aggregating the node features using the operation

$$\varphi_{\text{GAT}} = \text{LN}\left(\sum_{i \in \mathcal{N}} v_i\right).$$

Utilizing GAT allows us to extract deeper spatial and temporal features of the interaction between agents.

9. Fundamental Information

9.1. Dataset

We evaluate our method using two real-world datasets, nuScenes [6] and Argoverse [9], which developed by Motional are publicly available and large-scale, catering to autonomous driving research. They offer 2D or 3D annotations of road agents, track IDs and high-definition map data.

nuScenes The dataset offers high-definition maps and trajectory data from 1,000 driving scenes in Boston and Singapore, areas noted for dense traffic and complex driving challenges. It comprises 245,414 trajectory instances, each a sequence of 2D coordinates over 8 seconds, sampled at 2Hz. The nuScenes benchmark requires predicting a target agent's 6-second future trajectory from a 2-second historical trajectory. The comprehensive dataset features approximately 1.4 million camera images, 390,000 LIDAR sweeps, 1.4 million RADAR sweeps, and 1.4 million object bounding boxes across 40,000 keyframes.

Argoverse The dataset facilitates research in 3D tracking and motion forecasting for autonomous vehicles. Originating from select areas in Miami and Pittsburgh, it includes 113 scenes with 3D tracking annotations, featuring 324,557 significant vehicle trajectories derived from over 1,000 hours of driving. The forecasting component of Argoverse provides agent trajectories and high-definition maps, requiring the prediction of a target vehicle's future trajectory for the next 3 seconds, based on its past trajectory over two seconds, sampled at 10Hz. The dataset encompasses 333K real-world driving sequences, primarily at intersections or within dense traffic, each focusing on one target vehicle for trajectory prediction.

9.2. Experiment Detail

In this model, we set all embedding dimensions to 32 (including trajectory encoder and motif encoder). We report the results after 100-epoch training on 8 NVIDIA Tesla K80 GPUs. In this section, we present a selection of parameters utilized in the Spatial-Temporal Motif Matrix (STMM), Adaptive Spatial Interaction, Adaptive Temporal Interaction, and Graph Attention Network (GAT).

Details in STMM. As depicted in Figure 3 , the motif matrix is initially computed for each time slice by leveraging the historical trajectories of the ego vehicle and its neighboring vehicles. Subsequently, the motif matrix in the spatial dimension is obtained using the influence radius r chosen from the set $\{10, 20, 30, 40, 50\}$. With a sampling frequency of 2Hz and a historical trajectory duration of 2 seconds, the temporal dimension has a total length of 4. Consequently, a total of 20 motif matrices are obtained by multiplying the dimensions in the temporal and spatial domains, resulting in 4×5.

Details in ASI. Along the temporal dimension, the 20 decoupled spatial-temporal motif matrices were summed up. Subsequently, each resulting summed motif matrix undergoes min-max normalization and is fed into an SPP-Net with 5 max-pooling scales (16×16, 8×8, 4×4, 2×2, 1×1). Furthermore, motif matrix features were extracted using one fully connected (FC) layer and one layer normalization (LN) layer, converting the 341-dimensional motif features into fixed 64-dimensional features. Importantly, if the size of the motif matrix was smaller than the max-pooling kernel in the SPP-Net, a 2x upsampling operation was employed to expand the dimensions. Using the extracted 64-dimensional motif features, two separate mappings were performed to obtain the 'key' and 'value' components. Within the key-value pairs of the 5 motif matrices, the 'query' is obtained by concatenating the 32-dimensional global feature extracted by PointNet and the 32-dimensional ego-vehicle feature, utilizing an FC layer. Subsequently, a multi-head self-attention operation with head$= 5$ is employed to extract additional spatial interaction features from the attention mechanism applied to the motif matrix.

Details in ATI. We summed up 20 decoupled spatial-temporal motif matrices along the spatial dimension, resulting in 4 motif matrices that serve as inputs to the ATI block. Similarly to the ASI block, these 4 motif matrices underwent SPP-Net, FC, and LN operations, resulting

Table 5. Ablation Study of Loss

A_*	L_{DIS}	L_{SEL}	L_{KLD}	ADE_1	FDE_1	ADE_5	FDE_5	ADE_{10}	FDE_{10}
A_1	$\surd$	$\times$	$\times$	2.99	6.81	2.34	5.40	1.93	4.30
A_2	$\surd$	$\surd$	$\times$	2.52	5.61	2.52	5.81	2.32	5.23
A_3	$\surd$	$\times$	$\surd$	3.42	7.98	1.77	4.00	1.29	2.65
A_4	$\surd$	$\surd$	$\surd$	2.84	6.59	1.39	3.14	1.02	2.05

in a 64-dimensional motif feature. Subsequently, a fixed 256-dimensional feature for all agents was extracted using PointNet and concatenated with the 64-dimensional motif feature. Afterwards, an FC layer is utilized to map this feature to the weight of the motif matrix at each time step. To ensure that the weights sum up to 1, a softmax operation is employed to normalize them.

Details in GAT. A three-layer Graph Attention Network (GAT) was utilized, where the initial node features of the GAT network for each agent comprised the x and y coordinates, heading, and velocity of the agent's original trajectory points, forming a 4-dimensional vector and resulting in 16 dimensions. Furthermore, a 32-dimensional feature generated by the trajectory encoder, along with the 32-dimensional features produced by the ASI block, were incorporated to capture the interaction between the vehicle and motif matrix. Consequently, the total dimension of the node features amounted to 80, obtained by summing the individual dimensions of 16, 32, and 32. In GAT, the edge feature consists of three dimensions: the edge feature represents the weighted sum of the motif matrices generated by the ATI block, along with the coordinate differences between the end and start points of the edge. The node features are fixed at 80 dimensions for each layer of GAT. Following the final GAT layer, a single FC layer and one LN layer are employed to map the 80-dimensional feature to a more compact 32-dimensional representation.

10. Ablation Study of Loss

In this paper, we utilize three loss as follows:

$$Loss = L_{\text{DIS}} + \lambda_1 L_{\text{SEL}} + \lambda_2 L_{\text{KLD}},$$

where parameters (λ_1, λ_2) is $(1, 0.5)$ as previously mentioned. For estimate different effects for the performance of the proposed DAMM model, which can understand its function deeply in prediction task. Specifically, evaluate the effectiveness of the loss functions. A comparison provide an understanding of their effect on the DAMM model so that their performance in terms of accuracy, stability and robustness can be assessed. Understand the effect of the loss function on the behaviour of the model. Different loss functions affect the training of the model and the resulting output. This experiment can reveal the impact of loss functions on the patterns and behaviour of model learning and provide insight into the role and trade-offs of loss functions in model optimisation.

The table 5 shows the specific design of three loss, and the percent is the higher value than fundamental L^2 norm loss DIS.

Effects of DIS Loss. The existence of L_{DIS} is justified by its role as a fundamental loss function for quantifying the disparity between predicted trajectories and ground truth. When considering only L_{DIS}, the model primarily emphasizes minimizing the distance error between the predicted trajectory and the ground truth. As a result, the model overly prioritizes trajectory accuracy and underutilizes other valuable information, such as interaction between agents. The results analysis indicates elevated errors in ADE and FDE compared to the baseline. Notably, the FDE_5 value stands at 5.4, representing an increase of 2.1 over the baseline value of 3.28. This outcome stems from an excessive focus on trajectory-specific details, resulting in less accurate predictions in complex traffic scenarios and potential limitations in generating diverse and long-term trajectories.

Effects of SEL Loss. In this analysis, we specifically examine experiments A_2 and A_3 to investigate the impact of L_{SEL}. In experiment A_2, we employ a binary cross-entropy operation to assist the DAMM model in capturing selected features and incorporating selection behavior during trajectory prediction. This approach leads to the generation of more diverse trajectories, improving the prediction results, increasing accuracy, and enabling increased diversity. Remarkably, in comparison to A_1, the results indicate significant enhancements, with a specific increase of a 15.72% improvement in ADE_1. In experiment A_3, where L_{SEL} is absent, the inclusion of L_{KLD} allows the incorporation of the prior distribution of latent variables in the CVAE framework. This facilitates the generation of diverse trajectories by the model. However, without the loss function related to the selection of trajectory samples, the model may generate trajectory samples that inadequately adapt to specific target scenarios. As a result, the ADE_{10} metric shows a 2.58 de-

crease compared to the A_2 experiment.

Effects of KLD Loss. The loss function L_{KLD} employs the Kullback-Leibler divergence operation within the CVAE framework. In experiments A_1 and A_2, we exclude the CVAE framework. L_{KLD} enhances the generative diversity and robustness of the model. Thus, in experiment A_3, the model achieves a balance between the accuracy and diversity of trajectories, allowing for the generation of diverse predictions. The incorporation of the prior network for trajectory prediction during the inference stage in the CVAE architecture leads to a significant rise in ADE and FDE by 24.36% and 25.93% respectively when $K = 5$, and 35.50% and 38.37% when $K = 10$ compared to A_1. However, since the CVAE framework and KLD loss primarily enhance diversity, A_3 is not as effective as A_1 when $K = 1$.

Overall Loss Function. Through the simultaneous utilization of three loss functions, the model achieves a comprehensive integration of accurate geometric shapes of trajectories, selection behavior on complex roads, and diversity in generated results across various scenarios. The utilization of all three loss functions enables the model to collectively consider these factors, leading to trajectory prediction outcomes that exhibit improved accuracy, diversity, and robustness. In comparison to the A_1 experiment, all metrics demonstrate improvements, resulting in an overall average increase of 31.54% in ADE and 32.47% in FDE.

11. Broader Impact

Trajectory prediction plays a crucial role in various applications, such as autonomous driving and adaptive cruise control (ACC). In autonomous driving, trajectory prediction assists vehicles in anticipating the movements of other vehicles and pedestrians, enabling them to make informed decisions and prevent collisions. Similarly, in ACC, trajectory prediction enables the adjustment of vehicle speed and ensures a safe distance is maintained from other vehicles. By considering heterogeneous traffic flow density, our method enhances the accuracy and reliability of trajectory predictions, contributing to the development of safer and more efficient transportation systems. This technology finds applications in diverse transportation scenarios, encompassing urban and suburban areas, highways, and intersections. Moreover, it has the potential to facilitate the advancement of autonomous vehicles, thereby enhancing mobility and accessibility for individuals unable to drive or with limited mobility.

12. Limitation

This paper proposes the use of the motif matrix to predict the trajectory and direction of the agent, facilitating the extraction of significant dynamic features through the analysis of their variations. Owing to its complexity, it necessitates a substantial investment of time and computing resources. In this model, we select 3-node subgraphs by counting the number of triangles. More complex subgraphs, involving 4 or more nodes, have the potential to capture interactions from diverse perspectives and find application in various scenes, which can be explored in future research.

13. Visualization

Due to space constraints, the number of visualized prediction results provided in the main text is limited. Furthermore, we present additional visualized prediction results for various scenarios with trajectory counts K of 1, 5 and 10. We categorize the scenes into four parts based on density and lane difficulty: density (high and low) and lanes (complex and simple). "Density" refers to the density of neighboring agents. The lane degree indicates the scenario, with "complex lanes" referring to corners, intersections, and T-junctions, while "simple lanes" typically features straight roads with multiple lanes. In order to improve the clarity of the visualization results, we have created a dedicated page to present the specific outcomes. Here, we provide an explanation of the figure within the caption.

In conclusion, considering various combinations of scenarios and adjusting the number of generated trajectories leads to obtaining more diverse, accurate, and comprehensive vehicle trajectory prediction results. By increasing the number of trajectories, prediction diversity and realism are enhanced, while analyzing different scenarios aids in adapting to the diversity and complexity of real-world traffic environments. These advantages contribute to enhancing the performance and applicability of the model.

- $K = 1$

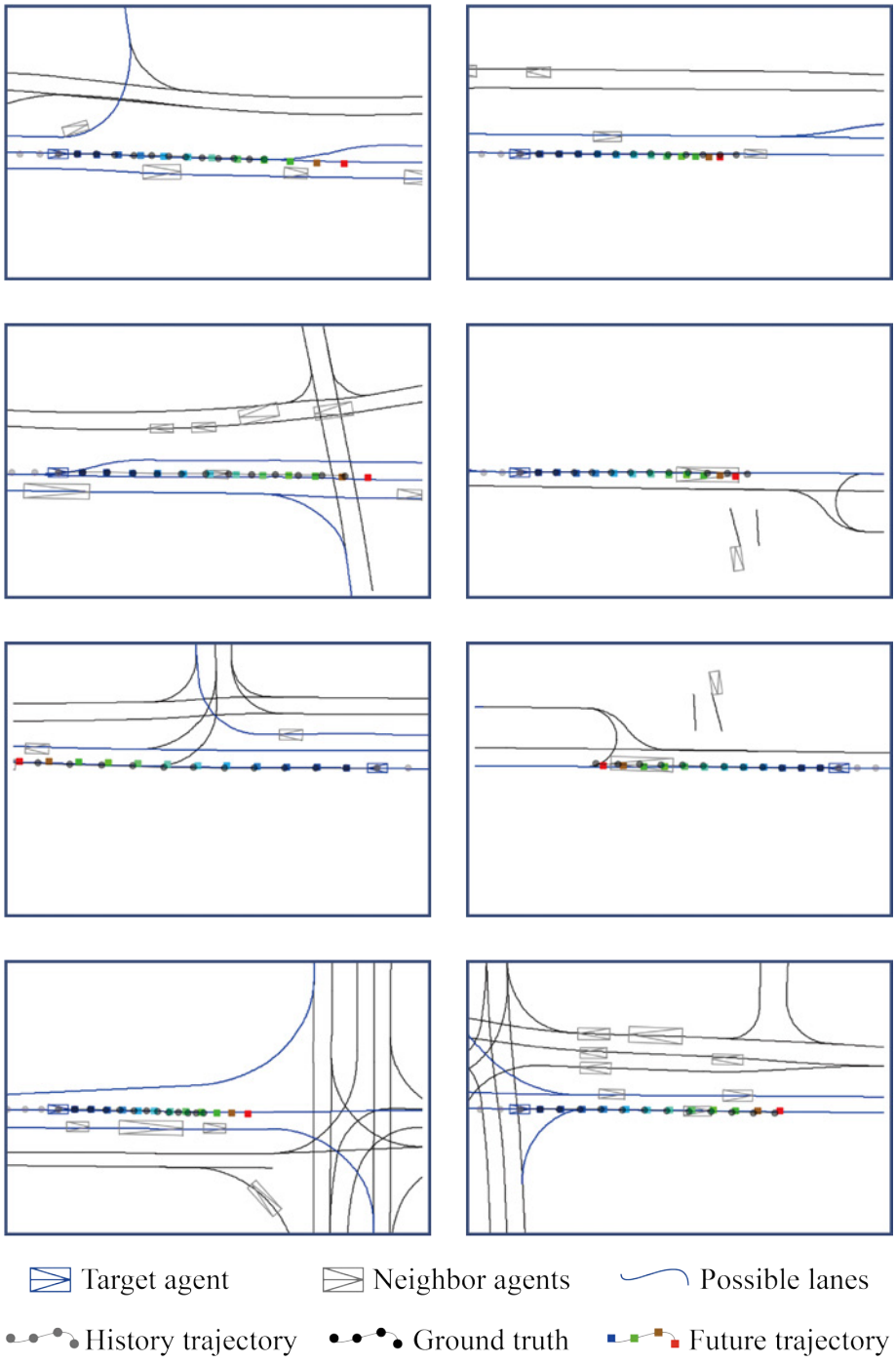

Figure 9. High density scenarios generated with $K = 1$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. In scenarios with a high density of neighboring agents, generating a single trajectory can provide a fundamental understanding of the primary behavioral patterns, enabling a quick prediction of the overall trend and motion direction of the agents, including behaviors such as following and overtaking.

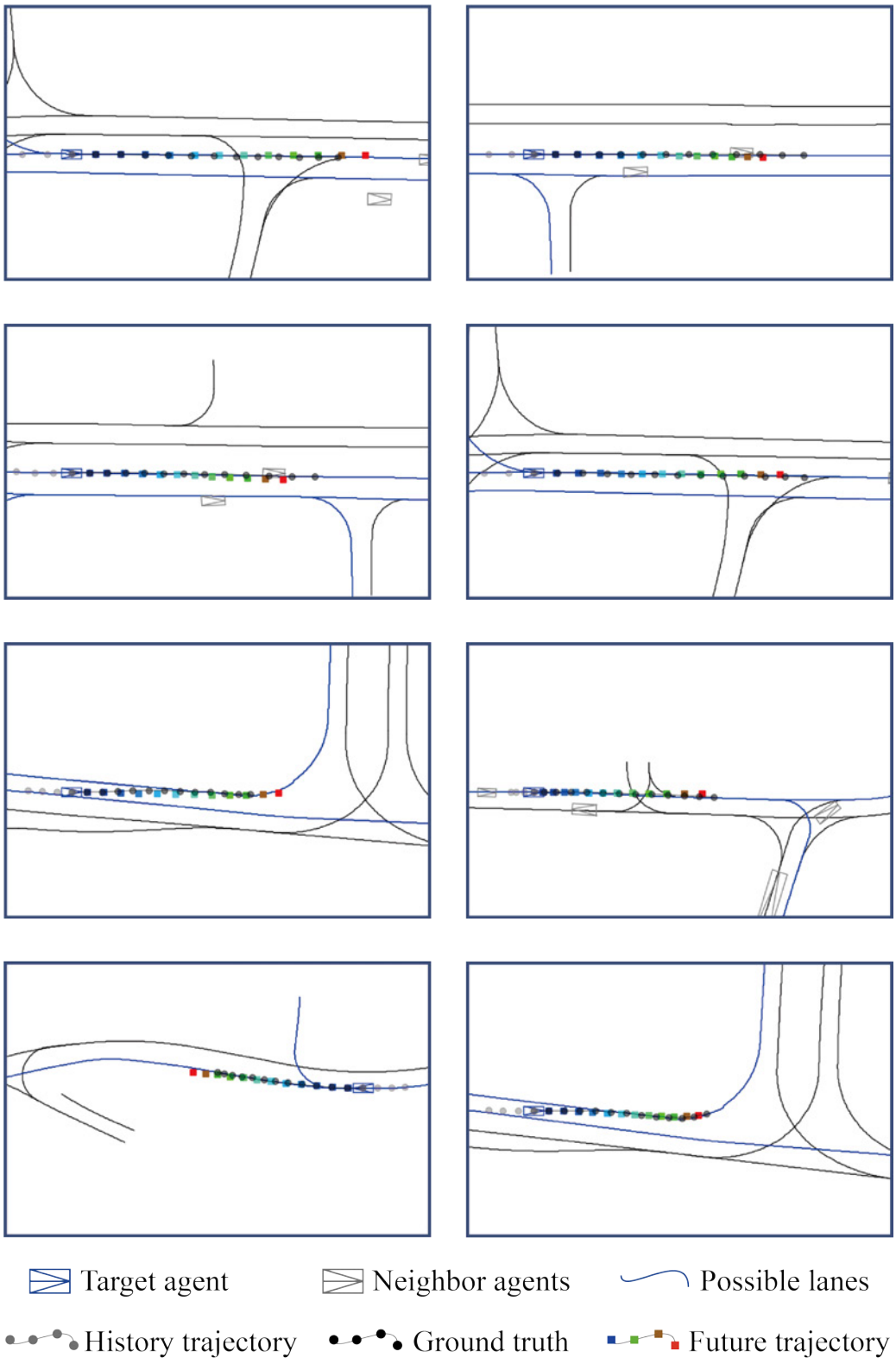

Figure 10. Low density scenarios generated with $K = 1$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. In sparse traffic environments, generating a single trajectory helps understand the agent's own motion patterns and path selection. With a lower density of neighbors, the potential for further interaction decreases, leading to increased accuracy in the generated trajectories.

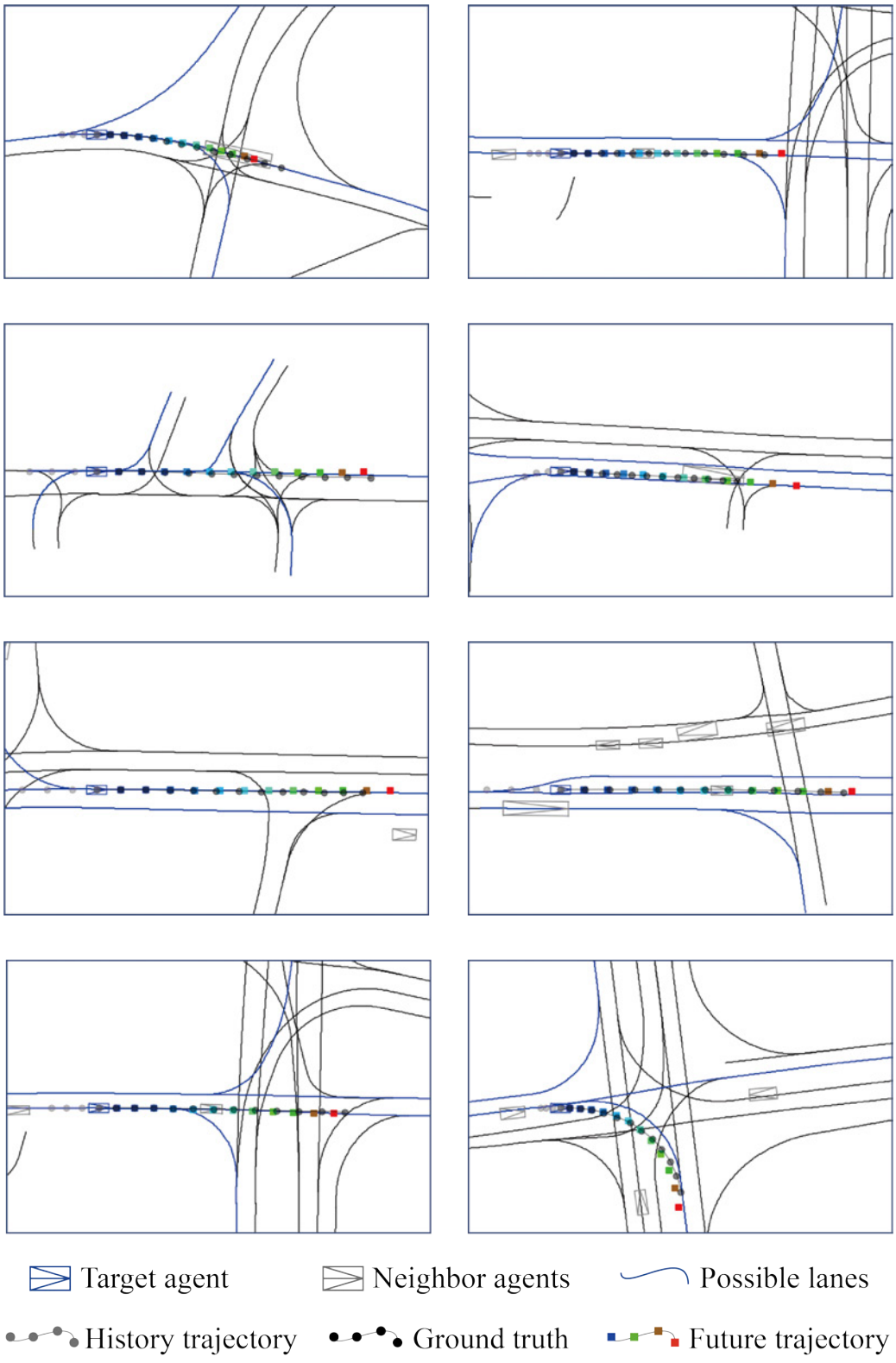

Figure 11. Complex lanes scenarios generated with $K = 1$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. In complex lane scenarios, generating a single trajectory can initially identify major features such as lane changes, turns, and intersections, offering an understanding of the overall trend.

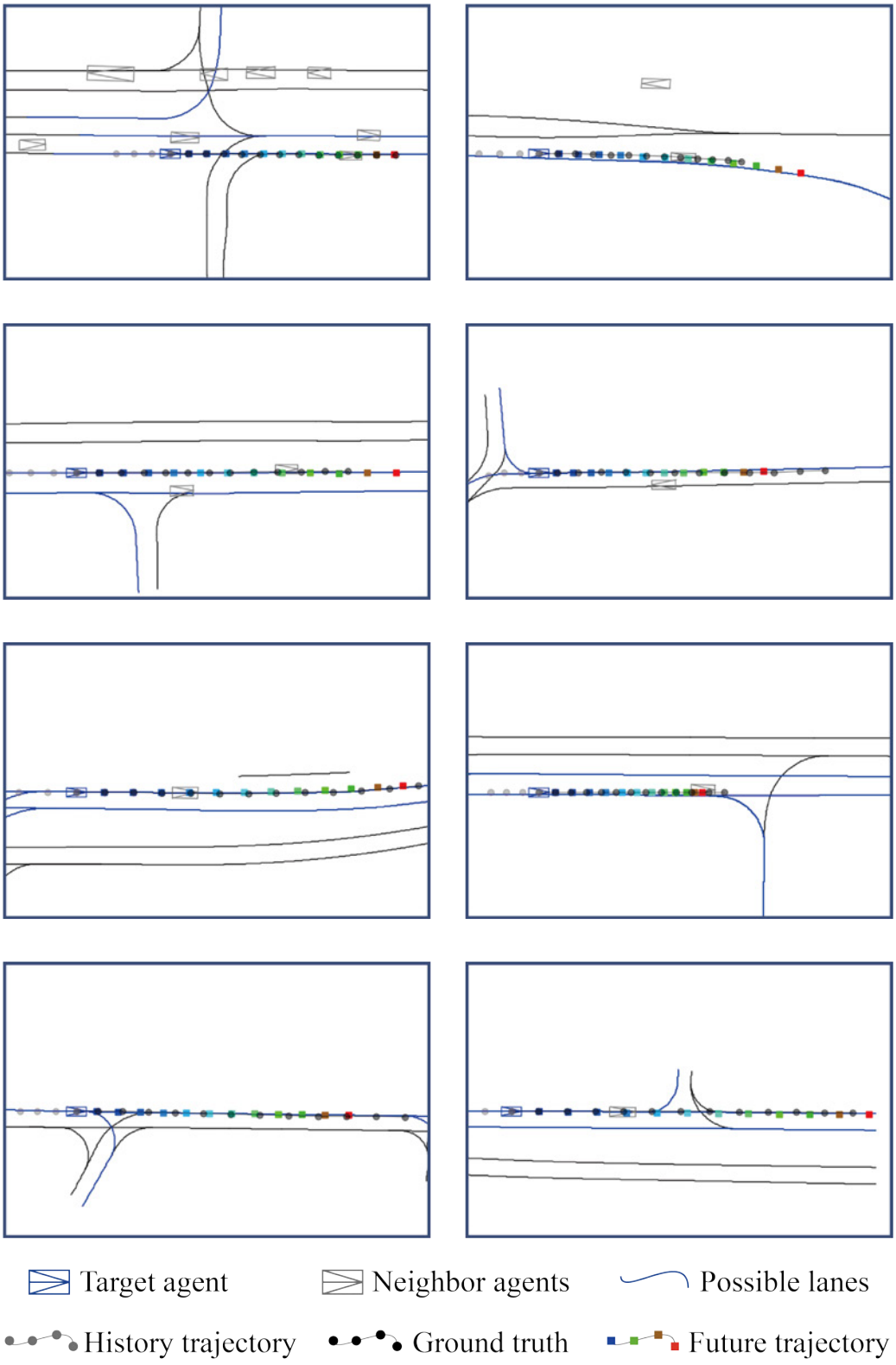

Figure 12. Simple lanes scenarios generated with $K = 1$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. In simple lane scenarios, generating a single trajectory can encompass most of the behavioral patterns, offering insights into the fundamental path selection and speed adjustment of the agents.

- $K = 5$

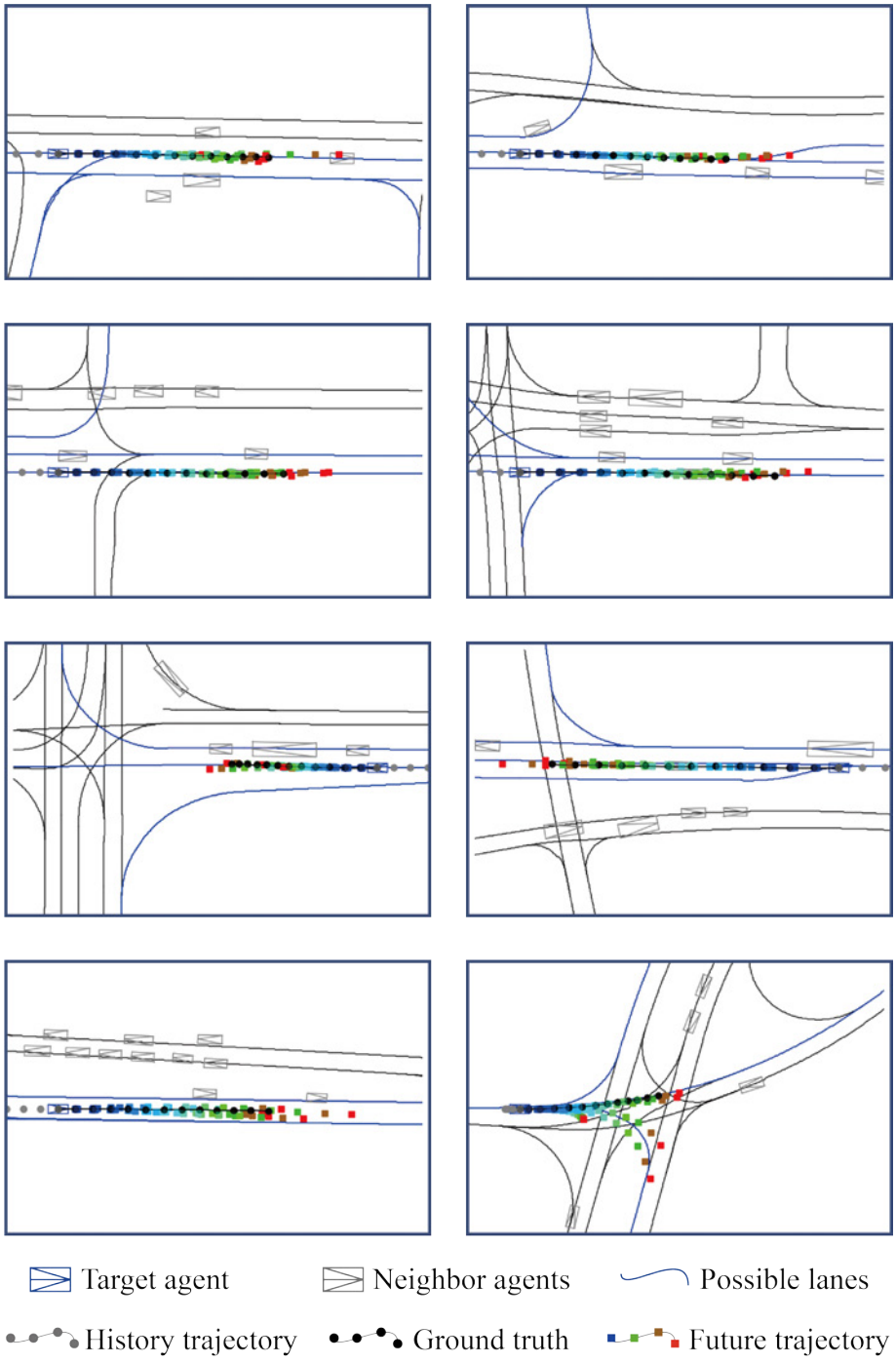

Figure 13. High density scenarios generated with $K = 5$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to five enables a better capture of the diverse interaction behaviors among neighboring agents in high-density scenarios. This facilitates a more comprehensive observation, capturing variations in behavior patterns like following and overtaking, consequently enhancing the diversity and authenticity of predictions. Partial overlaps indicate shared locations among individual segments.

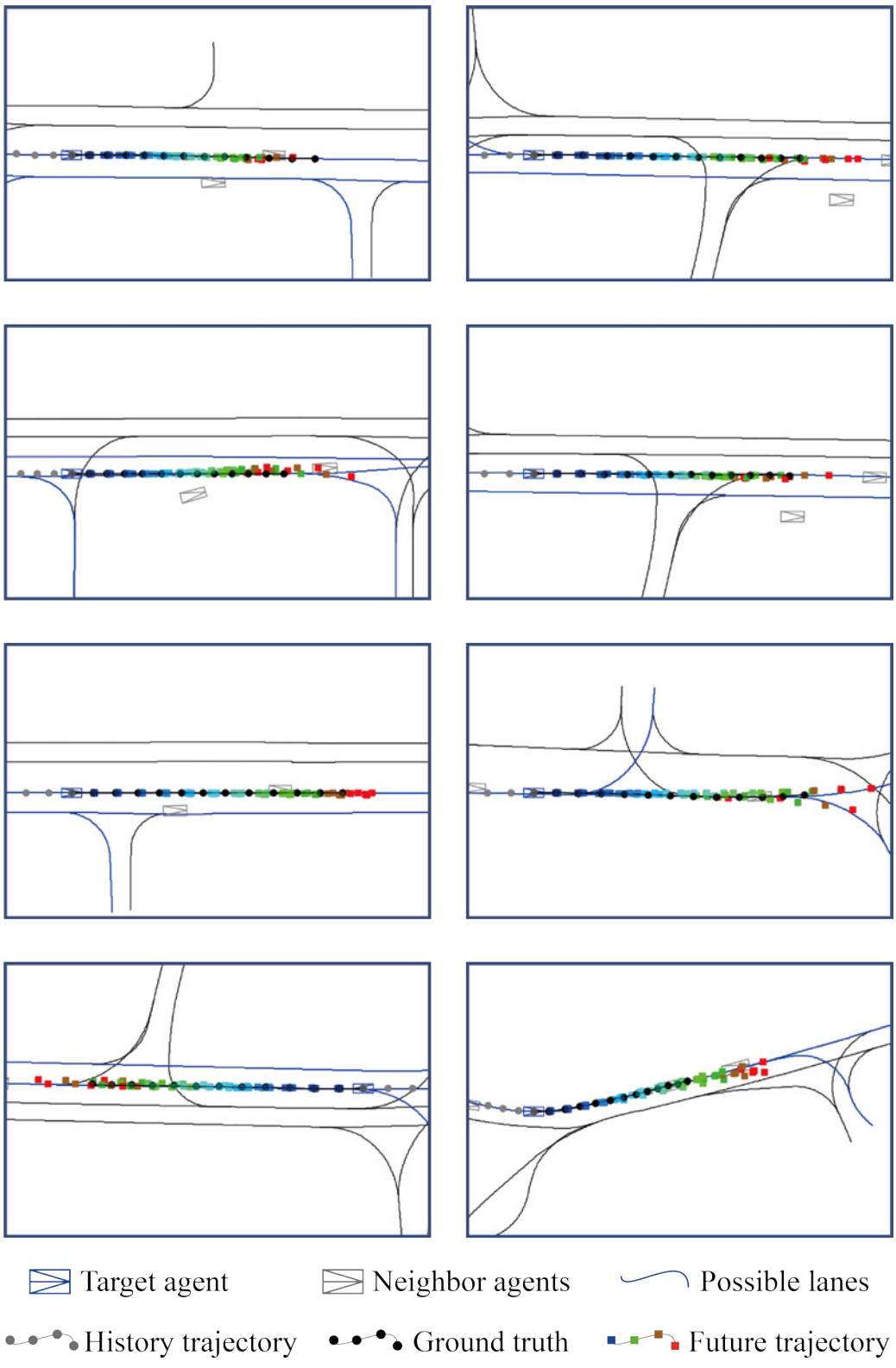

Figure 14. Low density scenarios generated with $K = 5$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to five better illustrates the diversity of the agents' behaviors, encompassing variations in speed, path selection, and other aspects. This enhances the realism and accuracy of the prediction results, particularly in scenarios with lower neighbor density.

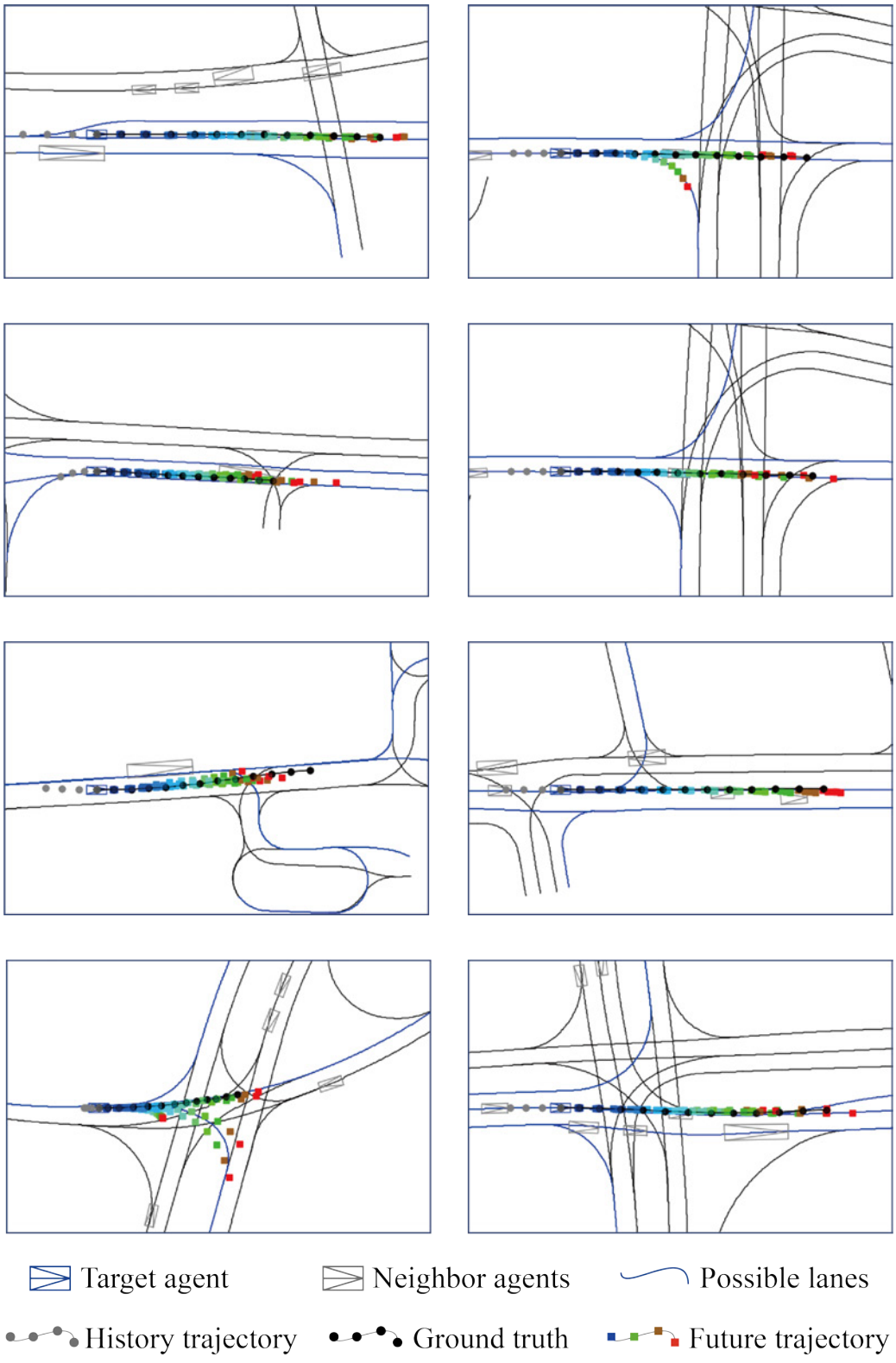

Figure 15. Complex lanes scenarios generated with $K = 5$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Generating five trajectories allows for a comprehensive capture of the adaptability and decision-making ability of agents in complex scenarios. The observation of more complex traffic markings and interactions between agents enhances the diversity and authenticity of predictions.

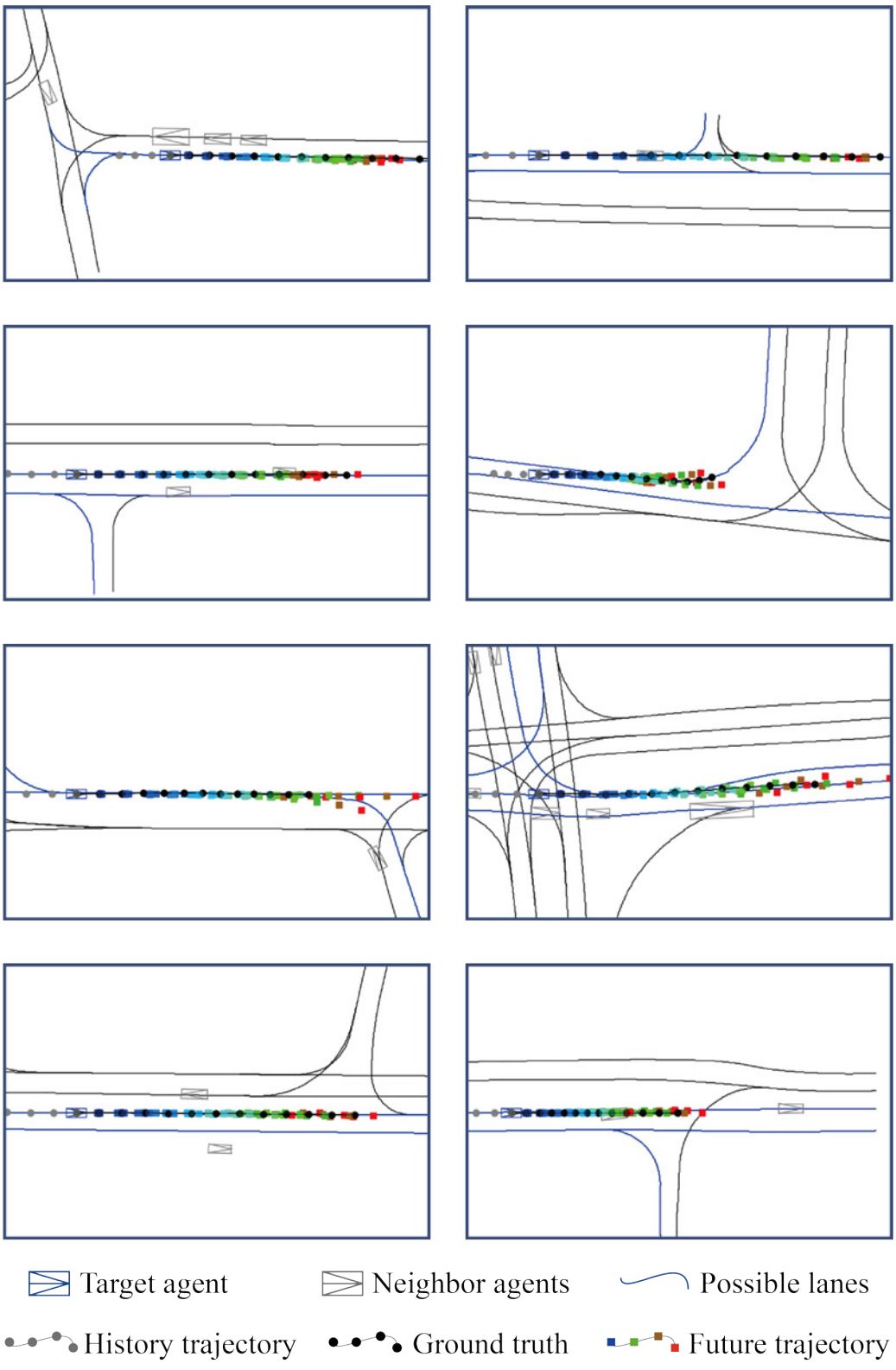

Figure 16. Simple lanes scenarios generated with $K = 5$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to five better showcases the diverse behaviors of agents in simple lane scenarios, capturing a wider range of speed variations, fine-tuning, and slight changes in path selection.

- $K = 10$

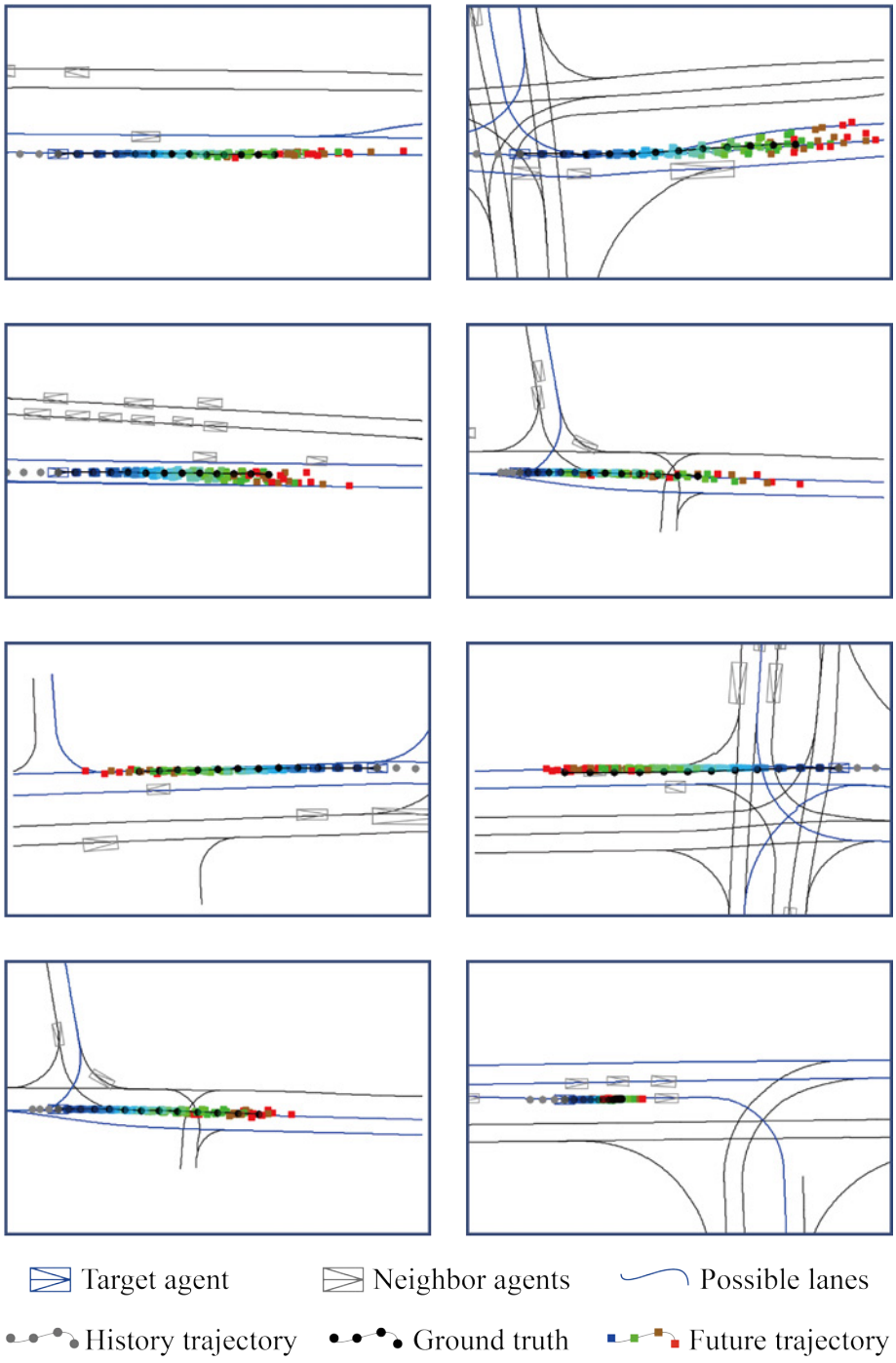

Figure 17. High density scenarios generated with $K = 10$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to ten further comprehensively reflects the diversity of complex interactive behaviors among neighboring agents. This leads to more accurate and comprehensive prediction results, better simulating the complex interactions between agents in real-world scenarios.

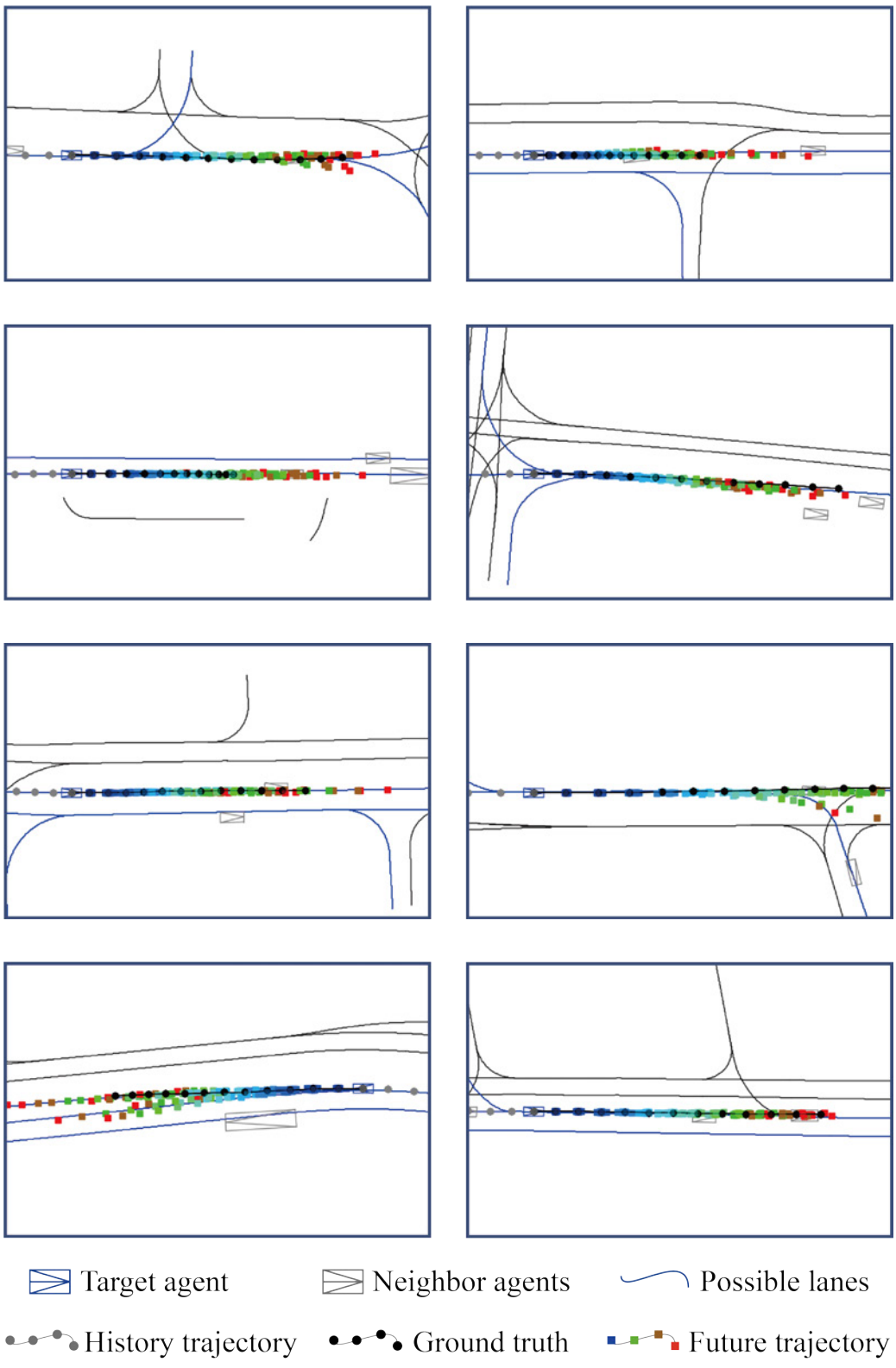

Figure 18. Low density scenarios generated with $K = 10$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to ten further enhances the observation of diversity in agents' own behaviors. This results in more comprehensive prediction results, encompassing variations in their speed, acceleration, path selection, and other aspects.

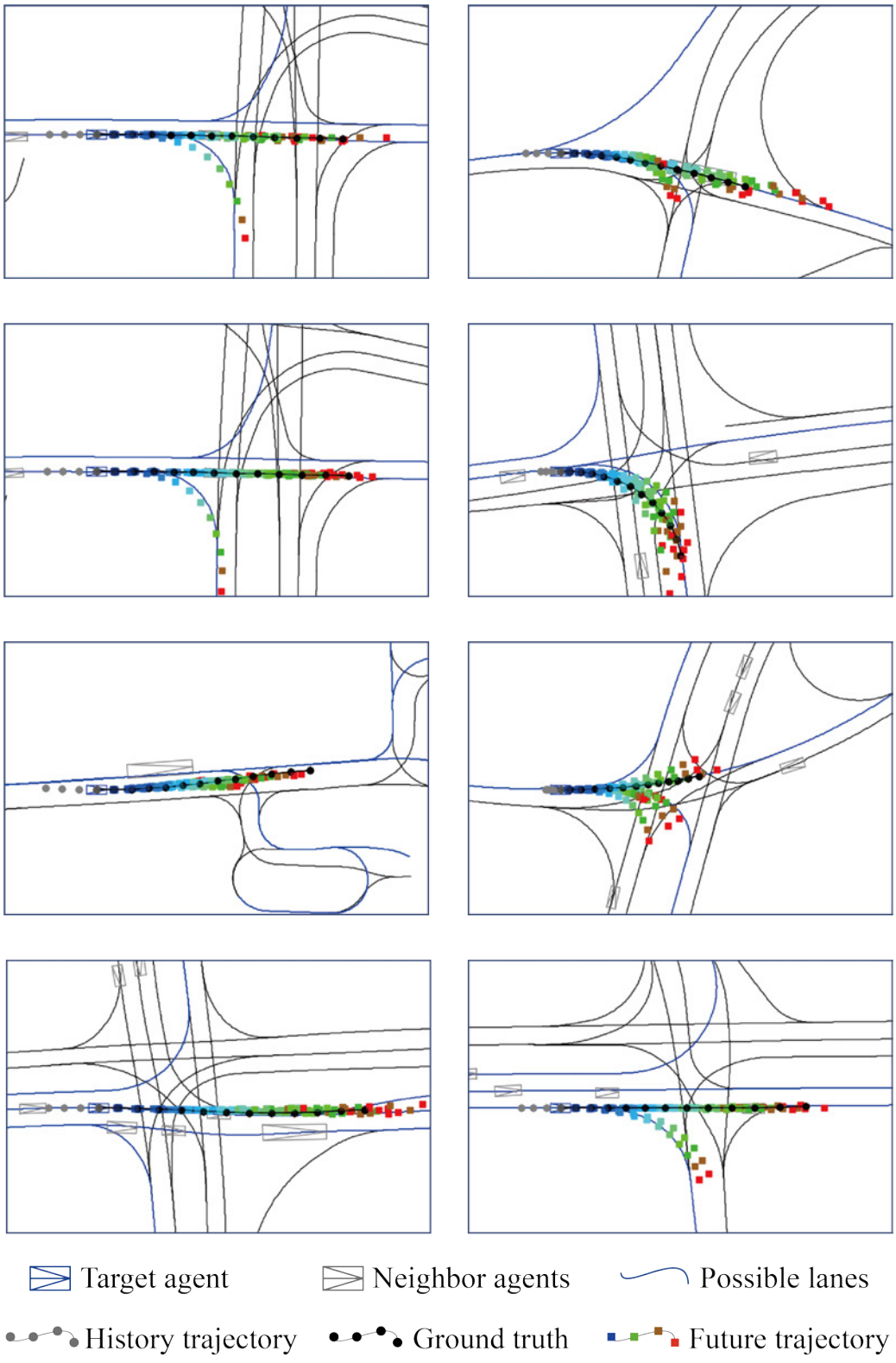

Figure 19. Complex lanes scenarios generated with $K = 10$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to ten allows for a more comprehensive consideration of the diverse behaviors and decision-making of agents in complex lane scenarios. This results in more reliable and comprehensive prediction results, better simulating complex traffic scenarios in real-world environments.

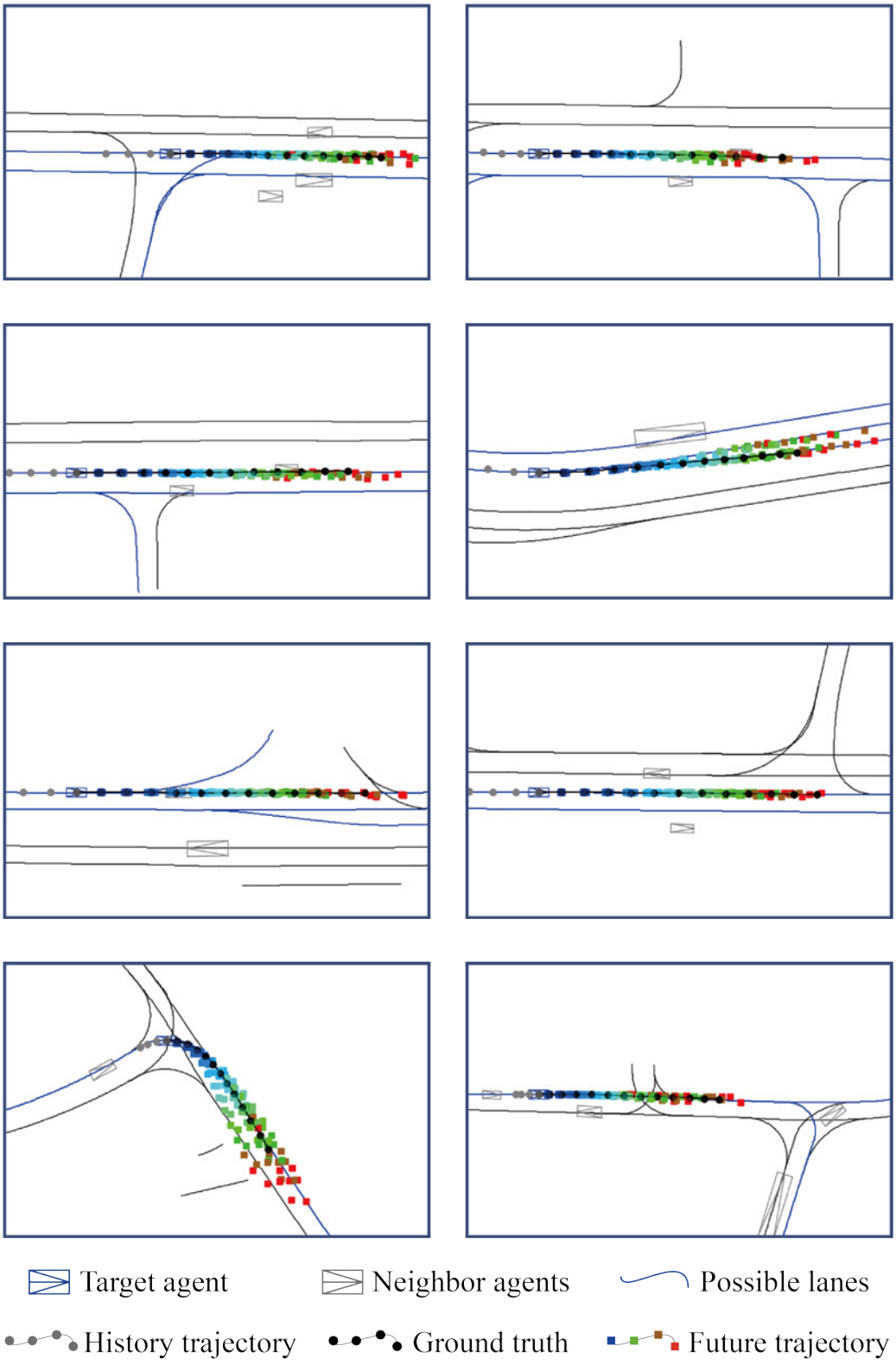

Figure 20. Simple lanes scenarios generated with $K = 10$. Each row depicts the predicted trajectory of the target agent (indicated by the light blue box) based on its history trajectories and neighboring agents (in the gray box). The predicted future trajectories are shown in various cases, with start points in blue and end points in red. The ground truth trajectory is represented in black for clarity. Increasing the number of trajectories to ten enables a more comprehensive consideration of the diverse behaviors of agents in simple lane scenarios. This provides more accurate prediction results and, by learning from previous processes, can simulate trajectory movements under upcoming road changes, thereby facilitating further decision-making.