Appendix

A Problem Background

A.1 Vehicle Trajectory Prediction

The methods for vehicle trajectory prediction can be broadly classified into four categories [100]. These include: (i) Physics-based methods: These employ vehicle dynamics or kinematics models, such as singles trajectory methods, Monte Carlo, and Kalman filtering methods [101, 102, 103, 104, 105, 106]. These methods are known for their conciseness, efficiency, and computational effectiveness. (ii) Classic machine learning: Unlike physics-based methods that rely on several physics models, classic machine learning approaches apply data-driven models and consider additional factors for predicting trajectories. Examples include the Hidden Markov Model, Dynamic Bayesian Network, and K-Nearest Neighbors [107, 108, 109, 110, 111]. However, these traditional methods are typically only suitable for simple prediction scenarios and short-term prediction tasks.

Recently, with the advancement of modern machine learning, vehicle trajectory prediction methods based on: (iii) Deep learning and (iv) Reinforcement learning become increasingly popular. These methods are capable of considering interaction-related factors, understanding high-dimensional complex policies, and adapting to more complex scenarios. Examples include Graph Convolutional Network, Graph Attention Network, Conditional Variational Auto Encoder, and reinforcement learning techniques such as Inverse Reinforcement Learning, Generative Adversarial Imitation Learning, and Deep IRL [112, 113, 83, 114, 115, 116, 117].

In summary, an increasing number of autonomous vehicle trials are utilizing deep learning or reinforcement learning methods to predict future vehicle trajectories. These approaches leverage expert demonstrations and extract interaction information from traffic participants and road conditions, considering a broader range of influencing factors.

A.2 Kolmogorov-Arnold Networks (KANs)

Hilbert's 13th problem [118] famously posits that it is impossible to solve general seventh-degree equations using only functions of two variables. Subsequent research by Kolmogorov et al.[36] has shown that any function involving multiple variables can be represented using a finite number of three-variable functions. Further studies as detailed by Arnol'd et al. [37], establish that even functions of just two variables are sufficient, as described in Theorem 2.1 presents significant for machine learning: learning a high-dimensional function essentially reduces to learning a limited number of one-dimensional basis functions $\psi(x)$ in Equation (2).

In reference [20], the authors introduce Kolmogorov-Arnold Networks (KANs), which are neural network applications based on Theorem 2.1. Unlike Multi-Layer Perceptrons (MLPs) that founded on the universal approximation theorem [119, 120, 121, 122], KANs feature learnable activation functions on what are traditionally referred to as "edges" (neurons) and they utilize fixed activation functions at what are typically called "nodes" (weights). Uniquely, each weight in KANs is replaced by a univariate function parametrized as a spline, meaning the network contains no linear weights whatsoever.

A variety of KANs are used across different tasks as noted in [123], such as solving ordinary differential equations [44, 124], image classification and reconstruction [125, 126], and time series forecasting [127, 128, 129], among others. These applications demonstrate competitive or superior performance in efficiency and predictive power compared to traditional models. However, to the best of our knowledge, we are the first to utilize KANs in vehicle trajectory prediction. This involves approximating and predicting trajectories for different driving styles, expanding the range of basis functions, and providing explanations for specific matches between functions and trajectories.

B Dataset Information

B.1 Datasets

nuScenes This dataset [67] 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 This dataset [68] 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.

Waymo This dataset [69] publicly to aid the research community in investigating a wide range of interesting aspects of machine perception and autonomous driving technology. This Dataset we use is the Motion part, with object trajectories and corresponding 3D maps for 103,354 segments. Given agents' tracks for the past 1 second on a corresponding map, predict the joint future positions of 2 interacting agents for 8 seconds into the future. The ground truth future data for the interactive test set is hidden from challenge participants. The validation sets contain the ground truth future data for use in model development. In addition, the test and validation sets provide 2 interacting object tracks in the scene to be predicted.

B.2 Metrics

We evaluate the predicted trajectory Y^t against the ground truth trajectory Y^t_{GT} using standard error-based metrics. Our DSA framework adopts the commonly used Average Displacement Error (ADE) and Final Displacement Error (FDE), as defined in [23]:

$$\text{ADE} = \frac{1}{T}\sum_{t=1}^{T} \|Y^t - Y^t_{\text{GT}}\|_{L^2}, \ \text{FDE} = \|Y^t - Y^t_{\text{GT}}\|_{L^2}.$$

Here, the superscript t denotes the current time step, and T refers to the total number of time steps in the prediction horizon. The metrics we use include ADE_k, FDE_k, minADE, minFDE, and b-minFDE. The subscript k indicates the Top-k most likely predicted future trajectories. The "min" variants (minADE and minFDE) compute the L^2 distance between Y^t_{GT} and the closest predicted trajectory Y^t across all generated samples, averaged over all agents. The b-minFDE metric extends minFDE by incorporating the Brier score [130], which evaluates the calibration of the predictive distribution. It is defined as the sum of the Brier score and minFDE.

C Proof of Bound

As we discuss in Section 3.3.2, Kolmogorov's theorem [131] provides error bounds by evaluating the absolute value of the function and the overall variation in the function value. This is illustrated as follows:

Theorem C.1 ***(Kolmogorov Theorem)*** *For $f \in C[a,b]$, there exists a polynomial p_n such that approximation error is bounded by:*

$$\|f - p_n\|_{L^\infty} \lesssim \left(\frac{\log n}{n}\right) V(f,[a,b]),$$

where $V(f,[a,b])$ denotes the total variation [8] of f over the interval.

From this theorem, we conclude that when $n > e$, increasing the degree n of p_n results in a smaller decrease in the theoretical upper bound on the approximation error.

However, in practice, considering computational cost and time, the value of n cannot be arbitrarily large. In this section, we provide three proofs corresponding to the three categories of drivers, demonstrating that under a given error limit δ, there exists a relationship between the minimum degree n and the components of the trajectory to be approximated, as described in Section 2.1 (i.e., position (x, y), velocity, and acceleration). For clarity, we use f to represent each continuous component with respect to t.

C.1 Conservative Drives: Bernstein Polynomial

In Section 3.2.1, we use the properties of Bernstein polynomials (B_n) for their uniform convergence to approximate the trajectories of conservative drivers, characterized by low speed and minimal motion changes. The minimum degree n of the B_n polynomial is obtained by:

Theorem C.2 *For all $\epsilon > 0$, if $\partial[B_n(f)] = n$, then for a given error limit δ with $0 < \epsilon \leqslant \delta \ll \infty$, then*

$$n \geqslant max \mid f''(\xi) \mid / 8\delta.$$

Proof: First, we calculate the error between $f(x)$ and $B_n(f)$

$$\begin{aligned}
&|f(x) - B_n(f)| \\
&= \left|\sum_{k=0}^{n}\left[f(x) - f\left(\frac{k}{n}\right)\right]\binom{n}{k}x^k(1-x)^{n-k}\right| \\
&\leqslant \left|\sum_{k=0}^{n}\left[-f''\left(\tfrac{k}{n}-x\right) - \frac{1}{2}f''\left(\frac{k}{n}-x\right)^2\right]\cdot P_B(k)\right| \qquad (4) \\
&= \frac{1}{2}f''(\xi)\sum_{k=0}^{n}\left(\frac{n}{k}-x\right)^2 P_B(k) \qquad \xi \in \left(x, \frac{k}{n}\right). \qquad (5)
\end{aligned}$$

In formula (4), $P_B(k) \triangleq C_n^k x^k (1-x)^{n-k}$. From Equation (5), we next proceed to prove

1. $\sum_k \left(\frac{n}{k} - x\right) \cdot P_B(k) = 0$,
2. $\sum_k \left(\frac{n}{k} - x\right)^2 \cdot P_B(k) = \frac{x}{n}(1-x)$.

For Equation 1, applying the Central Limit Theorem (CLT) as discussed in [132], we consider the total difference of weights, represented by $[(n/k) - x]$. Specifically, the probability p satisfies $p = x = k/n$. In this case, as described in [133], we have $E(k) = nx$. Thus, we can derive the expectation as follows:

$$E\left(\frac{k}{n}\right) = \frac{E(k)}{n} = x. \qquad (6)$$

Therefore, Equation 1 simplifies to $E(k/n) - x = 0$.

For Equation 2, we employ a similar method; here $(\frac{n}{k} - x)^2$ represents the squared difference in weights between $\frac{k}{n}$ and x, alternatively described as the deviation between observation and expectation. According to Equation (6),

$$D\left(\frac{k}{n}\right) = \frac{nx}{n^2} \cdot (1-x) = \frac{x(1-x)}{n}.$$

[8] **Total Variation** A measure of the total amount of variation in a function over a given interval $[a,b]$, which is defined by $\sup_{x \neq y} |f(x) - f(y)| \,/\, |x - y|$.

Equation 2 corresponds to the squared deviation $\left(\frac{n}{k}-x\right)^2$ based on weights $P_B(k)$. Moreover

$$\because\ E\left[\left(\frac{k}{n}-x\right)^2\right]=D\left(\frac{k}{n}\right),$$

$$\therefore (5)=\frac{1}{2}f''(\xi)\frac{x(1-x)}{n}\leqslant\frac{M_2}{8n},\ \text{with } M_2=\max_{\xi\in\mathbb{D}}f''(\xi).$$

Considering the error limit δ, we have:

$$\frac{M_2}{8n}<\delta\Rightarrow n\geqslant\frac{M_2}{8\delta}.$$

□

Theorem C.3 *If $f\in L^p[a,b]$, $B_n^\omega(f)\in[a,b]$ and $\partial\left[B_n^\omega(f)\right]=n$. For all $\epsilon>0$, $0<\epsilon\leqslant\delta\ll\infty$, δ is given error limitation, then:*

$$n\geqslant\frac{\max[|f''(\widetilde{\xi})\cdot\omega|]\cdot(b-a)^2}{8\delta},$$

where ω is the weights of weighted B_n polynomials $B_n^\omega(f)$.

Proof: Here the error is L^∞ norm, the definition of $B_n^\omega(f)$ is

$$B_n^\omega(f)=\sum_{k=0}^{n}f\left(\frac{k}{n}\right)\omega\left(\frac{k}{n}\right)\binom{n}{k}x^k(1-x)^{n-k}$$

Let $u=(x-a)\,/\,(b-a)$, then $u\in[0,1]$. So $\widetilde{M_2}$ is similar to Theorem C.2, here

$$\widetilde{M_2}=\max_{u\in[0,1]}\left|g''(u)\right|=(b-a)^2\left|f''(\xi)\right|,$$

where $g(u)=f\left(\frac{u-a}{b-a}\right)$. The next following prove is similar to Theorem C.2. □

C.2 Aggressive Drivers: Chebyshev Polynomial

In Section 3.2.2, aggressive drivers' trajectories are characterized by non-smooth, high-speed movements during motion changes. We use the Chebyshev polynomials T_n^c and their minimum-maximum error properties to approximate these trajectories. The minimum degree n of T_n^c polynomial is obtained as follows:

Theorem C.4 *For $f\in L^p[a,b]$ and a given error bound δ (where $0<\epsilon\leqslant\delta\ll\infty$), the condition $\partial\left[T_n^c(f)\right]=n$ is satisfied:*

$$n\geqslant\frac{1}{\omega^{-1}\left(f,\frac{\delta}{12}\right)},\tag{7}$$

where ω^{-1} is the inverse of the modulus of continuity for the function f.

To provide the proof of Theorem C.4, we first introduce the definition of the modulus of continuity and a lemma related to this proof.

Definition C.5 *(Modulus of Continuity in L^p Space) Let $f\in L^p[a,b], p\geqslant1$ and $0\leqslant m\leqslant b-a$. The modulus of continuity $\omega_p(m,f)$ is defined as:*

$$\begin{aligned}\omega_p(m)=&\omega_p(m,f)\\=&\sup_{0\leqslant h\leqslant m}\left(\int_a^{b-h}|f(x+h)-f(x)|^p\right)^{1/p}\end{aligned}$$

which represents the continuity norm for f over the interval $[a,b]$.

For T_n^c polynomials belonging to the $C_{2\pi}$ space [9], we use $E_n(f)$ to denote the deviation of the approximation of f by a trigonometric polynomial T_n of degree n, as follow:

$$E_n(f)=\inf_{\{T_n\}}\|f-T_n\|.$$

This deviation satisfies:

[9] $C_{2\pi}$ **Space** Let $f\in\mathbb{R}$ with period 2π. Define

$$\|\ f\ \|=\sup_{-\pi\leqslant x\leqslant\pi}|\ f(x)\ |.$$

We call the above set the $C_{2\pi}$ space

Lemma C.6 ***(Jackson** [134]) Let $f \in C_{2\pi}$, then for all $n \in \mathbb{N}$, the following inequality holds:*

$$E_n(f) \leqslant 12 \cdot \omega\left(f, \frac{1}{n}\right).$$

It is evident that $T_n^c \subseteq C_{2\pi}$. Based on Lemma C.6, we present the proof of Theorem C.4.

Proof: According to the definition of δ, we have $\| f - T_n \|_{L^\infty} < \delta$. Lemma C.6 provides the modulus of continuity under the L^∞ space, so we need to relate ω_p from Definition C.5 to ω in Lemma C.6, which relates the L^p and L^∞ norms:

$$\|g\|_{L^p([a,b-h])} \leq (b-a-h)^{\frac{1}{p}} \cdot \|g\|_{L^\infty([a,b-h])}.$$

For the function difference $g(x) = f(x+h) - f(x)$:

$$|f(x+h) - f(x)| \leq \omega(f,h), \text{ for } \forall x \in [a, b-h].$$

Therefore, the $\omega_p(m, f)$ is related to $\omega(f, h)$ as follows:

$$\begin{aligned} \omega_p(h,f) &\leqslant \left\{ \int_a^{b-h} [\omega(f,h)]^p \right\}^{1/p} \\ &= \omega(f,h) \cdot (b-a-h)^{\frac{1}{p}}. \end{aligned} \tag{8}$$

When $h \to 0$, Equation (8) can be approximated as:

$$\omega_p \leqslant \omega \cdot (b-a)^{\frac{1}{p}}.$$

According to Lemma C.6 and satisfy the error limit δ, s.t. $E_n(f) \leqslant \delta$, we have

$$\omega\left(f, \frac{1}{n}\right) \leqslant \frac{\delta}{12}.$$

To obtain the lower bound on n, since $\omega(f, h)$ is a nondecreasing function with respect to h, we take its inverse function $\omega^{-1}(f, y)$ as follows:

$$\frac{1}{n} \leqslant \omega^{-1}\left(f, \frac{\delta}{12}\right). \tag{9}$$

Finally, the lower bound for n can be derived from inequality (9) as:

$$n \geqslant \frac{1}{\omega^{-1}\left(f, \frac{\delta}{12}\right)}.$$

□

Lemma C.4 provides a minimum bound related to the value of the modulus of continuity. Furthermore, the Lipschitz continuity [135] of vehicle trajectories X_i and Y_i offers a more compact bound for inequality (7):

Corollary C.7 *The bound in inequality (7) is satisfied as follows:*

$$n \geqslant \frac{12L}{\delta}, \tag{10}$$

where L represents Lipschitz constant.

Proof: The proof of Corollary C.7 consists of two parts: (i) establishing Lipschitz continuity of vehicle trajectories and (ii) deriving Equation (10).

(i) Lipschitz Continuity of Vehicle Trajectories

To demonstrate the Lipschitz continuity of vehicle trajectories, it suffices to show that their state information (Section 2.1), including (x, y) position, velocity v and acceleration a, satisfies the Lipschitz condition (L-condition). Specifically, there exists a constant L, s.t. for any $x', x'' \in [a, b]$, the following holds:

$$\left|f\left(x'\right) - f\left(x''\right)\right| \leqslant L\left|x' - x''\right|. \tag{11}$$

According to the physical relationships among these states, if acceleration a satisfies L-condition, then by the boundedness theorem [136], the other states also satisfy it. Thus, we take a as an example, and similar arguments apply to the other states, quod erat demonstrandum.

According to [137, 138], the variation in vehicle acceleration is constrained by factors such as engine performance, vehicle weight, and braking system, which implies that the jerk $j(t)$ (the rate of change of acceleration over time)

cannot be physically infinite. Therefore, there exists a constant M_j, s.t. $|j(\tau)| \leqslant M_j$. For any $t_1, t_2 \in [a, b]$, the following holds:

$$|a(t_1) - a(t_1)| = \left| \int_{t_2}^{t_1} j(\tau) \right| \leqslant \int_{t_2}^{t_1} |j(\tau)| \leqslant \int_{t_2}^{t_1} M_j \, d\tau = M_j |t_1 - t_2| .$$

Therefore,Hence, the acceleration function $a(t)$ is Lipschitz continuous with the Lipschitz constant $L = M_j$.

(ii) Derivation of a more compact bound.

To obtain a tighter bound, we use the L-continuity property of f. From the continues of modulus, we have:

$$\omega(f, h) \leqslant Lh \Rightarrow \omega^{-1}(f, y) \geqslant \frac{y}{L}. \tag{12}$$

From Jackson's inequality in Lemma C.6, let $y = \delta/12$ in Equation (12). This ensures that the error remains below δ, with $\delta/12$ acting as a piecewise error threshold. Then Equation (12) becomes:

$$\omega^{-1}(f, y) \sim \omega^{-1}(f, \frac{\delta}{12}) \leqslant \frac{\delta}{12L}. \tag{13}$$

Combining inequalities (7) and (13), we obtain the more compact bound (10). □

C.3 Normal Drivers: Legendre Polynomial

In Section 3.2.3, we discuss that the speed and acceleration of normal drivers maintain an intermediate state between conservative and aggressive drivers. Their trajectories do not change as dramatically as aggressive drivers, nor are they so slow as to affect the flow of traffic. To approximate these trajectories, we use the Legendre polynomial L_n. The minimum degree n of L_n is obtain by:

Theorem C.8 *For all $\epsilon > 0$, if $\partial [L_n(f)] = n$, for a given error limit δ with $0 < \epsilon \leqslant \delta \ll \infty$, then*

$$n \geqslant \left(\frac{C_H}{\delta} \right)^{1/\alpha}, \tag{14}$$

where C_H is Hölder constant modulus of continuity and α represents Hölder exponent.

From Theorem C.8 we apply Jackson's inequality (Lemma C.6) to establish a relationship between the approximation and the modulus of continuity. This inequality also applies to continuous functions defined on the interval $[-1, 1]$ interval. To achieve the bound in Equation (14), we further use the Hölder continuous property [139] of vehicle trajectory. Similar to L-continues as defined in Equation (11), Hölder continuous (H-continuous) is define as follows:

Definition C.9 ***(Hölder continuous)*** *For a function f defines on interval I, if there exits a constant $C \in \mathbb{R}$, s.t. for $\forall z^{\prime, z^{\prime\prime}} \in I$:*

$$\left| f(z') - f(z'') \right| \leqslant C_H \left| z' - z'' \right|^{\alpha}, \ \alpha \in (0, 1],$$

then f is is said to be Hölder continuous of order α.

When $\alpha = 1$, H-continuous reduces to L-continuous. Reference [140, 141, 142, 143] analyze the Hölder continuity or related smoothness of vehicle trajectories and their states (position, velocity, and acceleration), either directly or indirectly by examining the smoothness of physical constraints and changes. Therefore, the bound in Equation (14) can similarly be derived from inequality (12):

$$\omega(f, h) \leqslant C_H h^{\alpha} \Rightarrow C_H \left(\frac{1}{n} \right)^{\alpha} \leqslant \delta.$$

Thus, we obtain a bound as expressed by inequality (14).

D Effects of Framework Sensitive

In Section 4, we evaluate our DSA framework from the perspective of polynomial p_n dimensions. Specifically, the p_n combination and degree of basis functions, with a focus on the approximation and prediction effectiveness of the model. In this section, we further analyze the robustness of the DSA framework. In particular, we investigate how varying traffic densities (Section D.1) and different road conditions (Section D.2) affect performance. All experiments in this analysis are conducted on the **nuScenes** dataset.

D.1 Traffic Density

Traffic density is closely related to vehicle speed and traffic flow, and significantly influences trajectory prediction due to varying interaction patterns among vehicles [144]. To clearly present the impact of traffic density, we divide the dataset into five levels based on the number of vehicles per unit area. We compare the performance of our DSA framework against the best-performing baseline with publicly available code: Context-Aware [81], as identified in Table 1. The comparison results are illustrated in Figure 5.

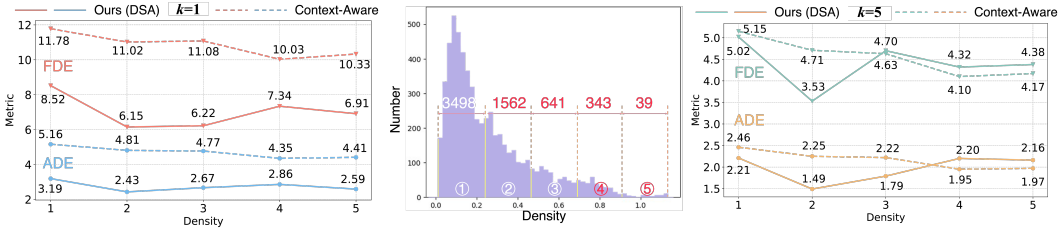


Figure 5: Comparison of trajectory prediction performance under different traffic densities with Context-Aware [81]. Our method is represented by solid lines, while Context-Aware is depicted using dashed lines on both sides. The middle subfigure shows the density distribution with circled numbers indicating the corresponding density levels.

Our DSA framework consistently outperforms the baseline across over 75% of the dataset. When averaged over the highest-density and most common case (density level 1 in the middle subfigure of Figure 5), our method achieves improvements of 1.97 in ADE_1 and 3.26 in FDE_1, as well as 0.25 in ADE_5 and 0.13 in FDE_5.

The most notable gains appear in density level 2, where DSA reduces prediction errors by 49.48% and 44.19% for $k = 1$, and by 33.78% and 25.05% for $k = 5$, compared to Context-Aware. Although our framework shows slightly lower performance in levels 4 and 5 when generating 5 trajectories, these cases together account for only 6.28% of the dataset.

D.2 Road Condition

Road structure significantly influences the motion patterns of agents navigating through urban or highway environments and is thus essential for accurate trajectory prediction [145, 11, 112]. Does road complexity increase the frequency of driving style changes, thereby making prediction more difficult? To investigate this question, we evaluate the performance of our DSA framework under different road conditions, as shown in Table 5.

Table 5: The b-minFDE in different road condition on the nuScenes dataset. The **best** results are highlighted.

Method Type	Stationary	Straight	Straight right	Straight left	Right U-turn	Right-turn	Left U-turn	Left-turn	All
MTR [146]	2.15	2.58	**4.85**	4.26	8.13	4.82	**5.17**	**4.85**	2.86
DSA	**2.03**	**2.48**	4.96	**4.17**	**8.11**	**4.77**	5.28	4.87	**2.75**

Our DSA framework achieves the lowest overall error of 2.75, improving upon the baseline by 3.85%. It outperforms the baseline in 6 out of 8 categories, including reductions in error for common scenarios such as Stationary (5.6%) and Straight (3.9%), as well as complex maneuvers like Straight-Left (2.1%), Right U-turn (0.25%), and Right Turn (1.0%).

Although MTR performs slightly better in Straight-Right and Left U-turn (by 0.11 in both cases), DSA matches or surpasses baseline performance in the most frequent and safety-critical trajectory types. These results demonstrate the robustness and adaptability of our framework across diverse road semantics, particularly in non-linear or discontinuous motion patterns.

E Detailed Description of the Algorithm

In Sections 3.3.1 and 3.3.2, we introduced the mechanisms for polynomial combination and degree adaptation. In this section, we provide a detailed description of the corresponding algorithms.

E.1 Polynomial Combination

To match each trajectory under various driving styles to a suitable polynomial combination, as analyzed in Section 3.2, we employ a Mixture of Experts model based on Top-K Gating (MoE-TopK) [63]. In this method, tunable Gaussian noise is added to the gating logits, and only the top K values are retained for expert selection.

Let us denote by $G(x)$ and $E_j(x)$ the output of the gating network and the output of the j-th expert network for a given trajectory X_i, for clearly we omit subscript i. The output z^{com} of the MoE module can be written as follows:

$$z^{\text{com}} = \sum_{j=1}^{3} G(X)_j E_j(x). \tag{15}$$

As shown in [21, 61, 62], kernel density estimation and latent variable analysis reveal that a driver's behavior evolves continuously across different situations. This implies that driving behavior can be viewed as a probabilistic mixture of weighted driving styles. Given that drivers may exhibit behaviors characteristic of multiple styles in dynamic scenes—such as when another agent suddenly appears—we adopt the Noisy Top-K Gating network [63] to capture this mixture behavior. This network activates only the top-k best-matching experts, enhancing responsiveness and specificity. Accordingly, this modification adjusts $G_*(\cdot)$ in Equation (15) to $\widetilde{G(x)}$, detailed as follows:

$$\widetilde{G_j(x)} = \text{Softmax}\left\{\text{KeepTopK}\left[H(X), k\right]\right\}, \text{with}$$
$$H(X) = (X \cdot W_{G_j}) + \text{SN}() \cdot \text{Sp}\left[\left(X \cdot W_{\text{noise}_j}\right)\right].$$

Here, "SN" and "Sp" denote Standard Normal [64] and Softplus [65] functions, respectively. The symbol W_* denotes the weight matrix corresponding to each subscript. The loss function is defined as follows:

$$L_{\text{MoE-K}} = w_{\text{load}} \cdot \text{CV}(\text{loads})^2, \tag{16}$$

where "load" refers to the importance values assigned to each driving style, with w_{load} representing the corresponding weight. "CV" stands for the Coefficient of Variation, which assesses the variability of these values. Equation (16) is a part of Loss in Section 4.1. This structure of the MoE model effectively recognizes the diversity of trajectories, allowing each expert to specialize in different features of driving styles.

E.2 Degree Adaptation

From Theorem 3.7, we understand that the accuracy of polynomial p_n approximation is directly influenced by the degree n of the p_n. However, adapting the degree of p_n poses a complex, non-convex, and combinatorial optimization challenge, as the relationship between prediction error and polynomial degree is not straightforward. This complexity often leads to the presence of multiple local optima.

To address this issue, we utilize the versatile Bayesian Optimization (BO) tool SMAC3 [66] for its robustness and flexibility, making it particularly suitable for optimizing low-dimensional and continuous functions (type: SMAC4BB), such as those found in vehicle trajectory prediction.

We treat the adaptive of polynomial degree as a hyperparameter optimization problem, using SMAC3 for BO, which leveraging Gaussian Processes with the Matérn kernel and the Expected Improvement acquisition function, iteratively searches the candidate degree set to minimize the loss function. Specifically, the degree n is treated as a hyperparameter optimization problem, aimed at minimizing the loss on validation data D_{val} of our model trained on training data D_{train}. This process can be formulated as follows:

$$n_{\text{SMAC}} \in \underset{n \in \mathbb{Z}^+}{\arg\min}\, c(n) = \underset{n \in \mathbb{Z}^+}{\arg\min}\, L\left(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}; n\right),$$

The hyperparameter optimization process targets the degree n_{SMAC}, which is defined as the optimal degree that achieves the least error for the corresponding basis function p_n. Here L denotes the loss function.

F Limitation and Discussion

Limitation We summarize existing open vehicle trajectory prediction datasets in Table 6, and observe that the maximum available trajectory duration is typically less than 10 seconds. Despite this limited time span, our framework based on three driving styles adapts well to such settings. We evaluate its performance on both short-term (3 seconds, Table 1) and long-term (8 seconds, Table 2) prediction tasks, achieving consistently strong or state-of-the-art results across all durations.

Table 6: Existing vehicle trajectory datasets. "His" and "Pre" represent the historical and predicted trajectory durations, respectively, while "Total" denotes the overall duration for each vehicle.

Datasets	Pub.	Collect Locations	His	Pre	Total
KITTI	2012 CVPR	Karlsruhe	2	4	6
Apolloscapes	2018 CVPR	Beijing, ShangHai and SHenZhen	3	3	6
nuScenes	2020 CVPR	Boston and Singapore	2	6	8
Argoverse	2019 CVPR	Miami and Pittsburgh	2	3	5
INTERACTION	2019 arXiv	China, Germany and Bulgaria	1	3	4
InD	2020 TIV	German	3.2	4.8	8
RounD	2020 ITSC	Aachen	2	4	6
HighD	2018 ITSC	German	2.8	2.8	5.6
Waymo	2020 CVPR	USA	1	8	9

However, in longer prediction horizons, the complexity of driving behavior increases, suggesting that three driving style categories may be insufficient to cover all possible scenarios. Moreover, trajectory patterns are often influenced by external factors, which can be categorized as either soft or strong conditions.

Soft conditions, such as weather, affect driver perception and reaction. For example, on sunny days, improved visibility may enhance drivers' responsiveness, leading to smoother and more stable trajectories. In contrast, adverse weather conditions such as fog, heavy rain, or snow can result in more abrupt or irregular driving behavior.

Similarly, strong conditions such as traffic signals or regulatory constraints also significantly influence vehicle trajectories. Unfortunately, most existing datasets lack labels for these contextual factors. We believe that incorporating such labels could further enhance prediction accuracy in future research.

Discussion For longer vehicle trajectories, we can improve our DSA framework from both practical and theoretical perspectives.

1. Incorporating more driving styles. Our current DSA framework utilizes three representative styles: Conservative, Aggressive and Normal (CAN), which reflect two behavioral extremes and an intermediate pattern. However, as the temporal length of each driver's trajectory increases, driving behaviors may exhibit greater variability. To capture these nuances, the model can be extended by defining or integrating additional driving styles. This would allow for a more fine-grained characterization of driver behavior and potentially lead to improved trajectory prediction accuracy.
2. Expanding the set of basis functions. As driving styles become more diverse and trajectory conditions more complex, a broader set of basis functions is required to effectively approximate and predict vehicle trajectories. Instead of relying on a single polynomial type, we can extend to a set of basis functions of the same class, such as orthogonal trigonometric polynomials. For example, to minimize the L^2 norm in modeling trajectories of normal drivers, or other intermediate states between conservative and aggressive behavior. It is beneficial to use a richer set of orthogonal polynomials that better match the dynamics of these nuanced driving styles.

G Future Work

In this paper, we focus on the characteristics of the individuals who generate the data (i.e., trajectories) and leverage the mathematical properties of basis functions to approximate these trajectories. This concept can be generalized and extended to other domains, such as:

- Other traffic participants. In addition to drivers, other agents in the traffic scene such as pedestrians and cyclists, also exhibit distinct behavioral characteristics. By modeling these characteristics, we can select appropriate basis functions tailored to each agent type, thereby improving the accuracy of their trajectory prediction.

- Multivariety Time Series Forecasting. Our framework can be extended to long-term forecasting tasks in domains such as weather prediction, energy consumption and electrocardiography. For example, one could model temperature and precipitation trends across different climate zones, analyze electricity usage patterns based on consumer behavior, or study heart rate dynamics as a function of individual health conditions.

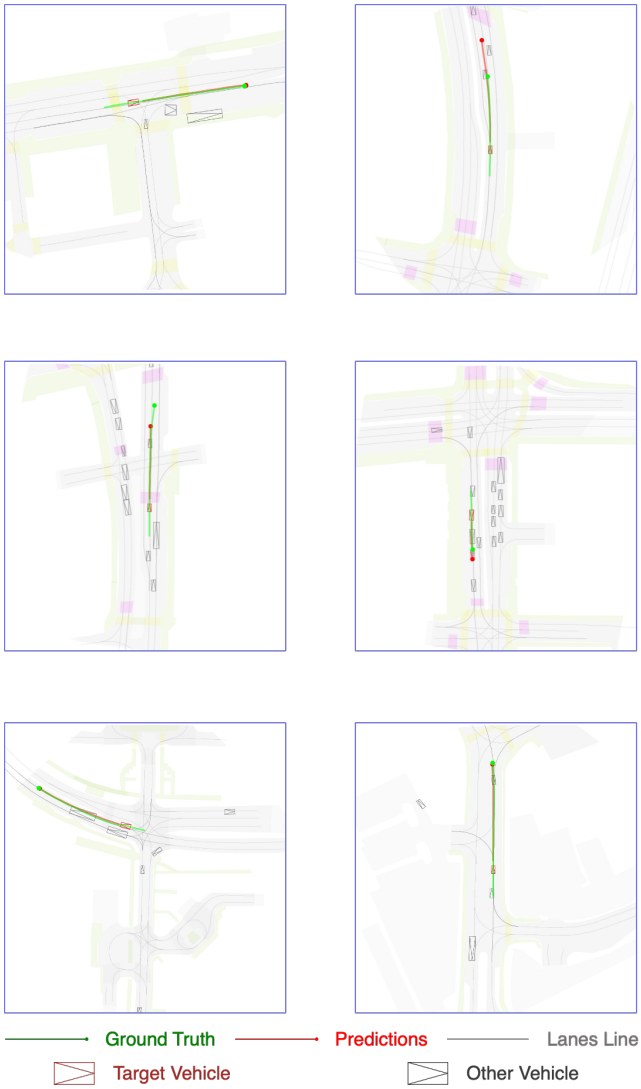
Additionally, by leveraging the core theoretical foundation (Theorem 2.2), we aim to construct models grounded in the physical characteristics or behavioral attributes of the data sources, thereby fully exploiting the inherent structure of the data itself.

H Visualization

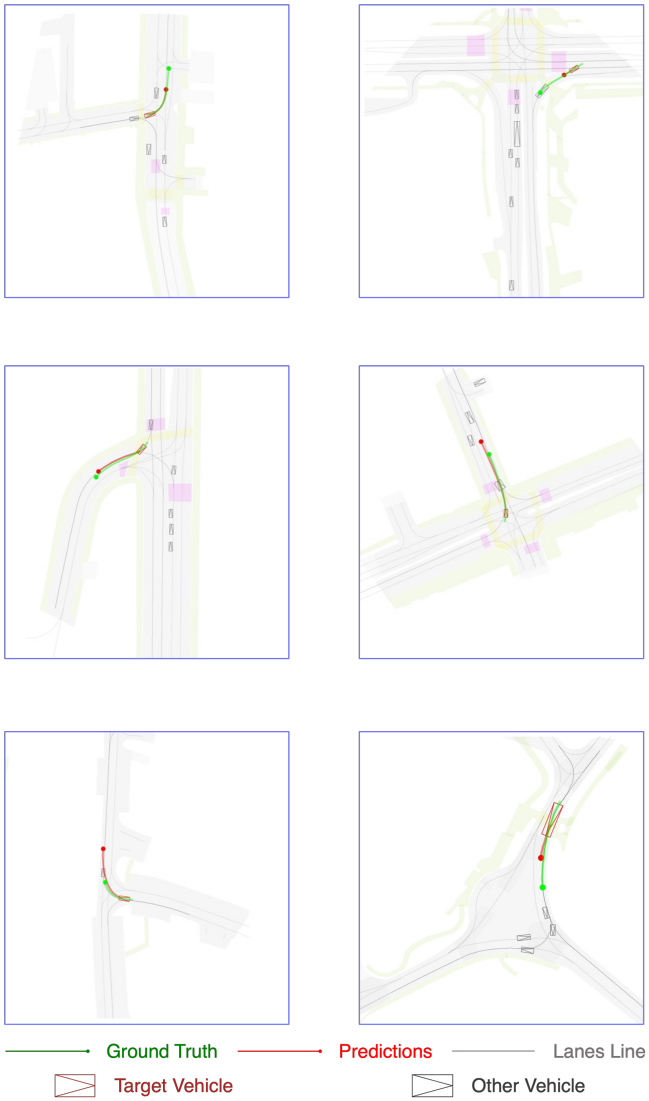
Due to space constraints, the number of visualized prediction results in the main text is limited. Here, we provide additional visualizations of predicted trajectories for various scenes, with generated trajectories $k = 1, 5, 10$. Each value of k is presented for both simple (e.g., straight roads) and complex scenes (e.g., turns conditions), showcasing different types of driving behavior.

To enhance the clarity of the visualization results, we present them on a dedicated page and reduce the background opacity to improve visual contrast. Specific outcomes are accompanied by detailed explanations provided in the corresponding figure captions. In summary, considering various scenario combinations and adjusting the number of generated trajectories lead to more diverse, accurate, and comprehensive vehicle trajectory predictions. Increasing the number of predicted trajectories improves prediction diversity and realism, while analyzing different scenarios helps adapt to the diversity and complexity of real-world traffic environments. These improvements contribute to making the model both more mathematically grounded and more adaptive.

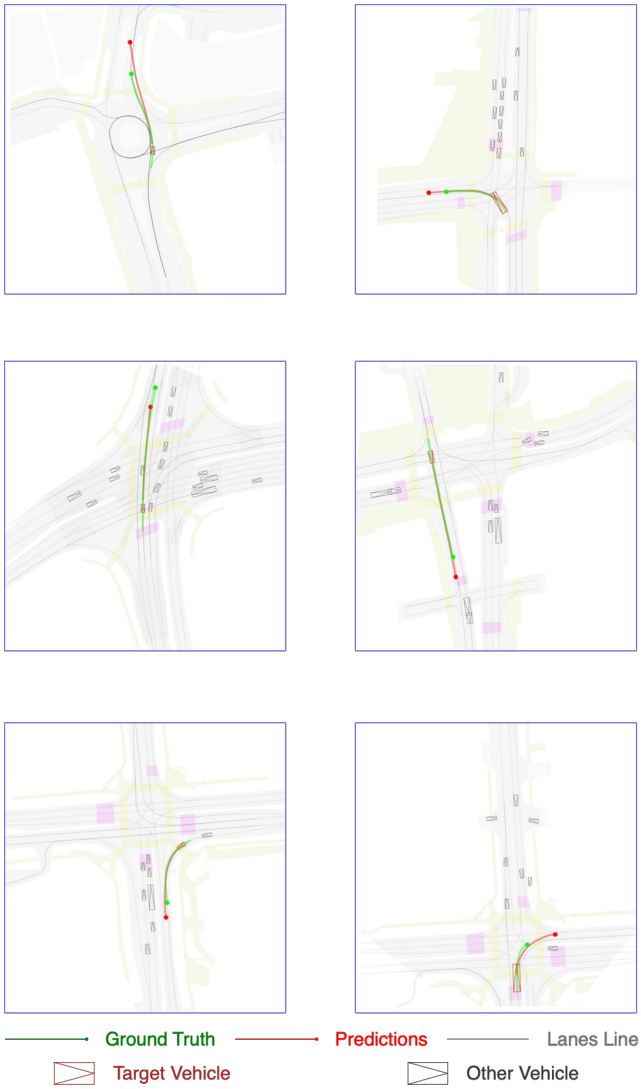
- $k = 1$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

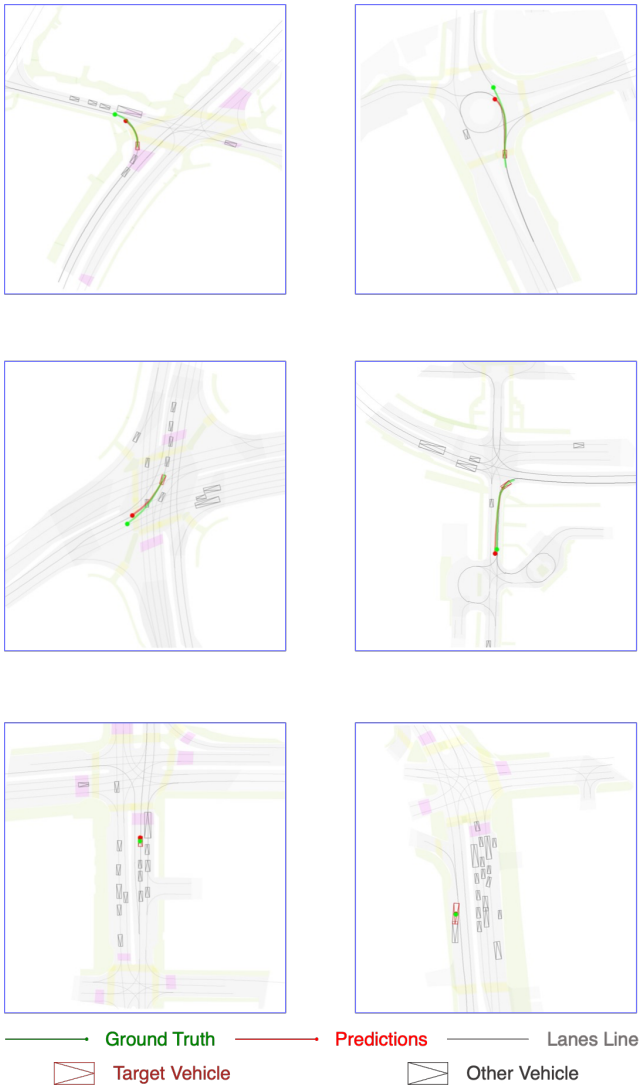
- $k = 1$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

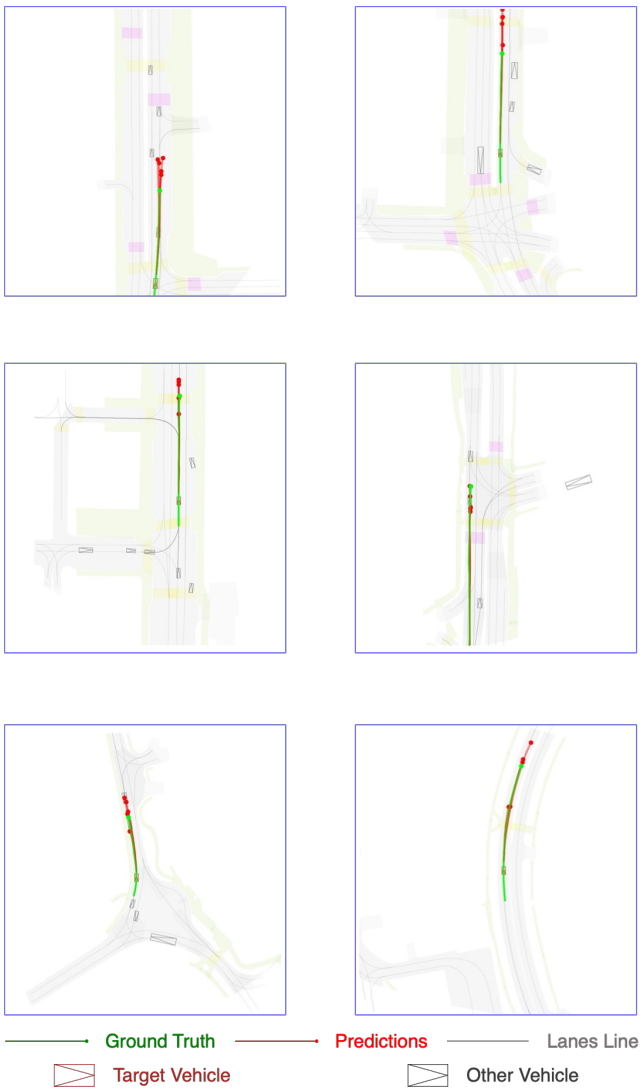
- $k = 1$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

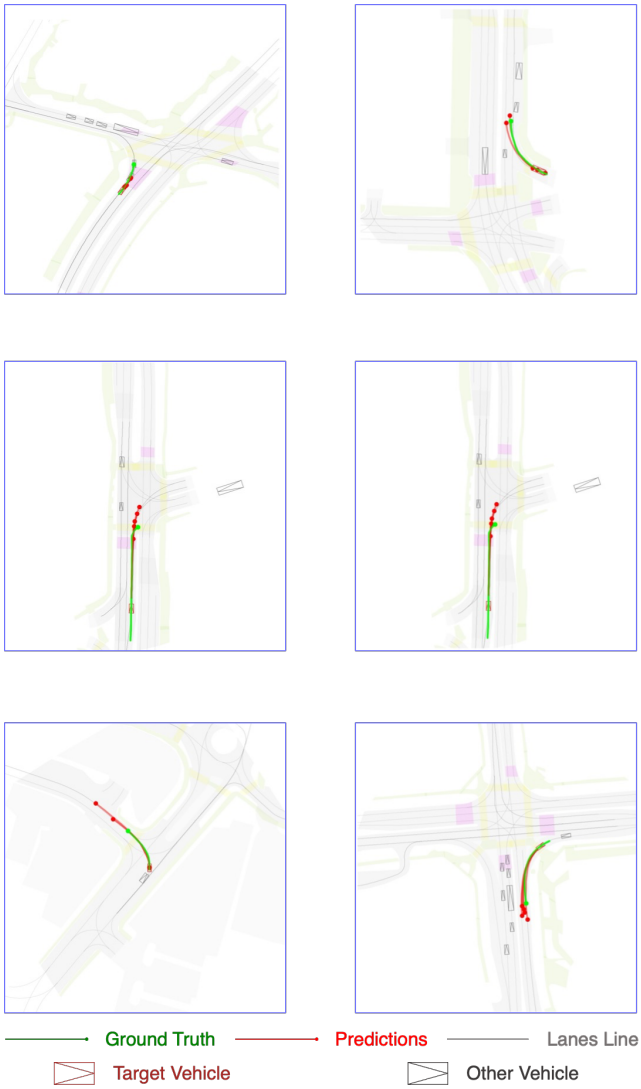
- $k = 1$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

- $k = 5$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

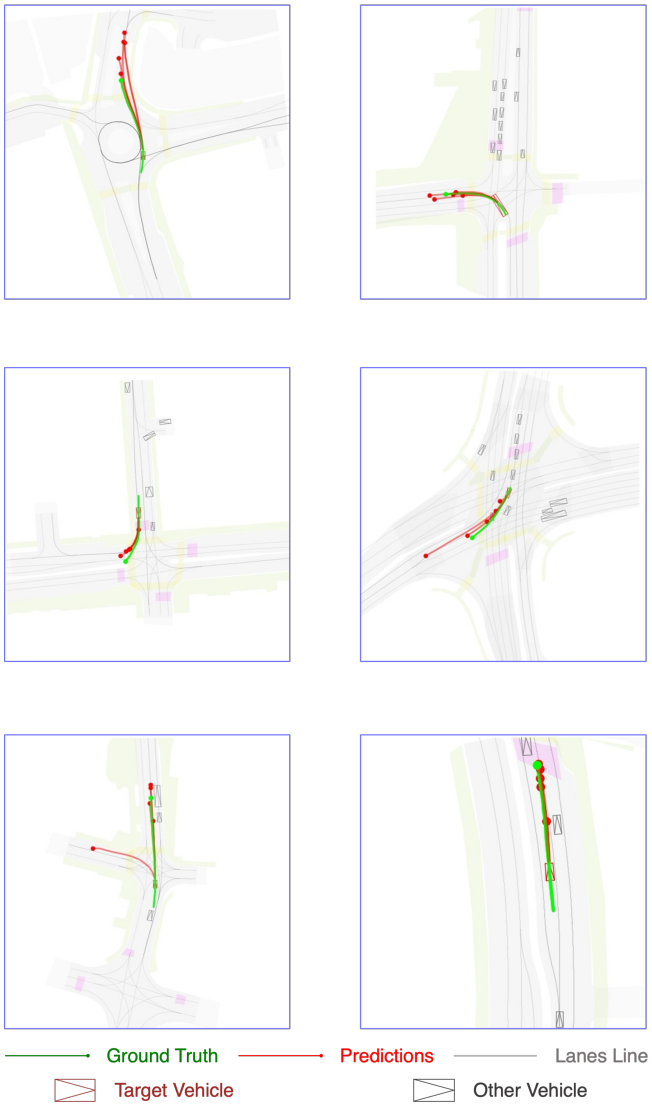
- $k = 5$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

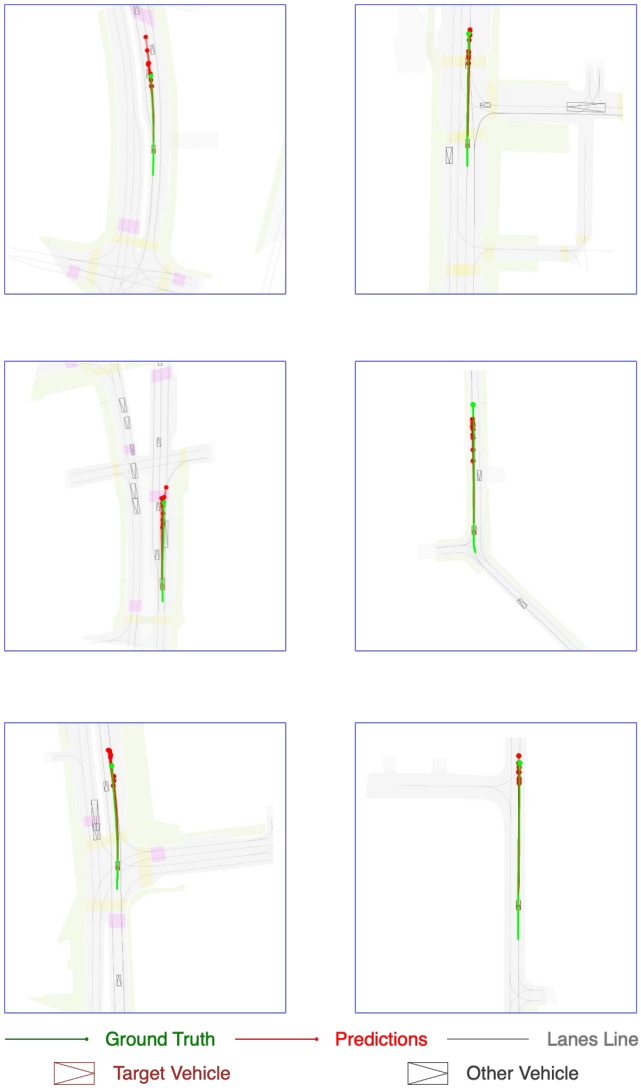
- $k = 5$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

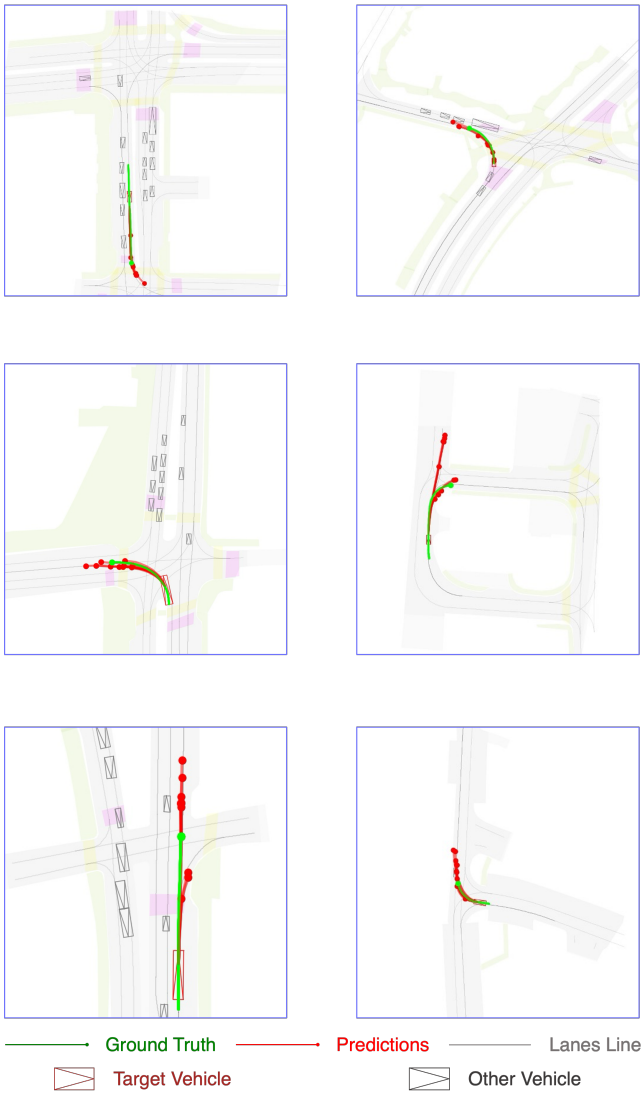
- $k = 5$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

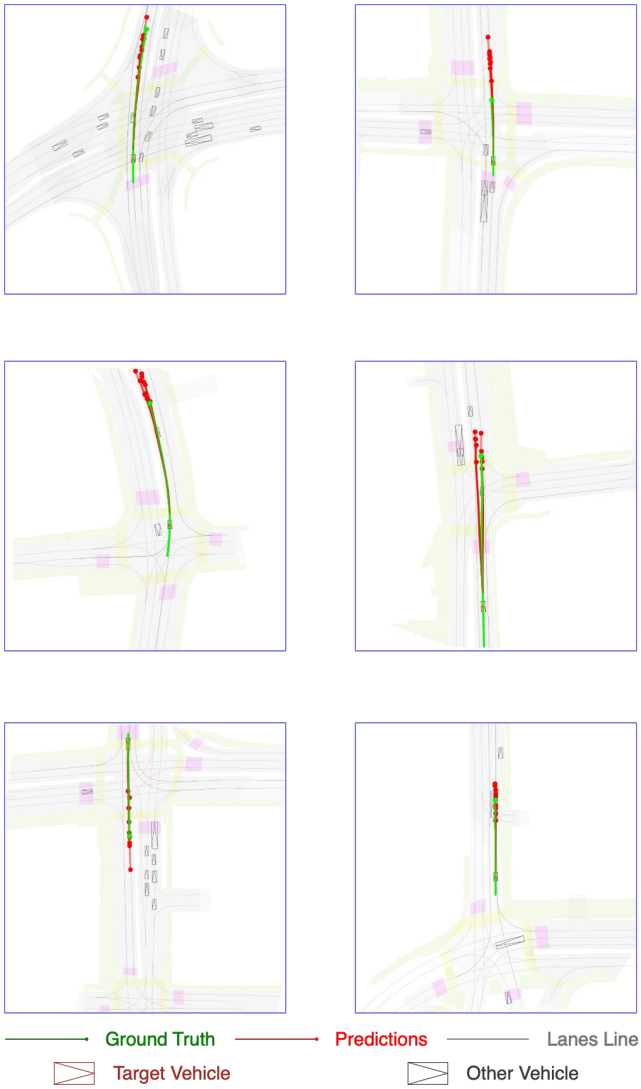
- $k = 10$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

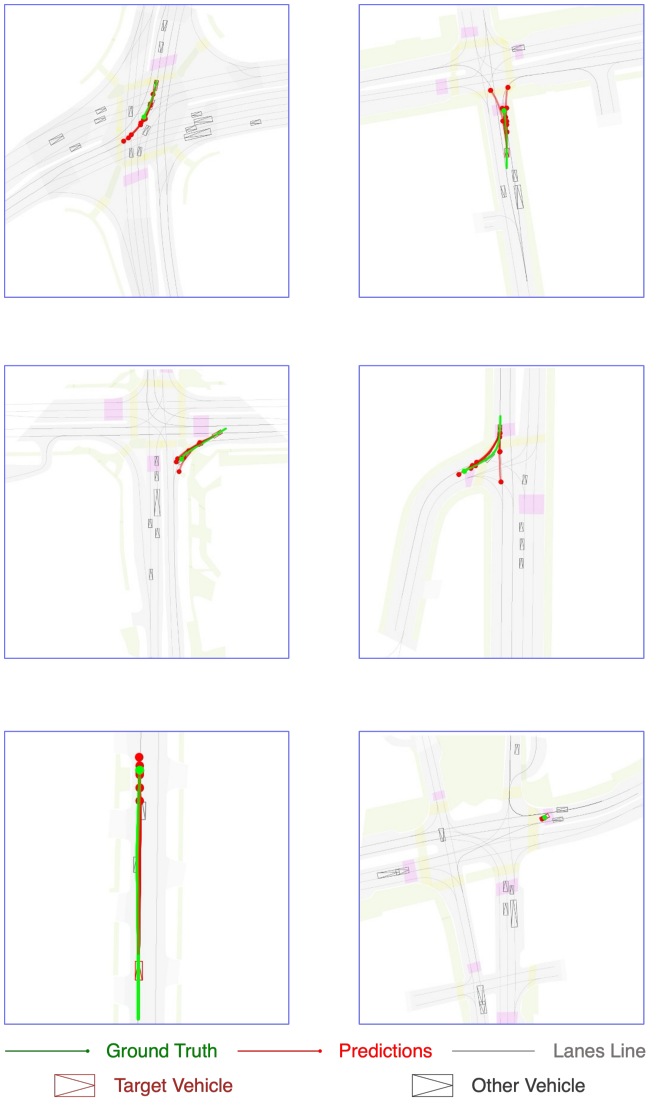
- $k = 10$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

- $k = 10$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle

- $k = 10$


Ground Truth
Predictions
Lanes Line
Target Vehicle
Other Vehicle