International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

Physics-Guided Interaction Sparsification for Efficient Ego Trajectory Prediction

[1]Abhishek Silwal, [2]Anku Jaiswal

[1]Department of Electronics and Computer Engineering, Pulchowk Campus (IOE, Tribhuvan University), Kathmandu, Nepal
[2]Assistant Professor, Department of Electronics and Computer Engineering, Pulchowk Campus (IOE, Tribhuvan University), Kathmandu, Nepal

***Abstract* - Agent Trajectory Prediction (ATP) predicts future motion of an agent (vehicle/individual) given the surrounding environment (pathways, pedestrians, vehicles) and the agent's previous motion history. This study aims to create a lightweight vehicle trajectory model that integrates driving lane information and physics-guided sparsification (i.e., inclusion of only the critical interacting nearby agents by selecting the select few based on physics-based neural network processing). The trajectory and other information was gathered using the nuScenes dataset and trained using different models to check the feasibility of both hard sparsification and physics-based gated learning using multi-layer perceptron (MLP) network. In this study, five models were created with different known architectures, among which the proposed model in this study implemented both physics-based gated learning using MLP and hard sparsification through selection of the top five critical neighboring agents based on the criticality score for interaction input. The models were compared on the basis of different evaluation metrics for accuracy, stability and safety. Qualitative analysis based on trajectory results and quantitative analysis based on robustness tests for missing or corrupt input was also performed to evaluate the models. Based on the results, the proposed model provided the most accurate and stable results for different scenarios, proving the effectiveness of combining gated learning with hard sparsification in creating an accurate and efficient ego trajectory model.**

Keywords: Trajectory Prediction, Sparsification, Ego Vehicle, nuScenes, Physics-guided.

I. INTRODUCTION

Agent Trajectory Prediction (ATP) implies predicting where an individual or a group of people or vehicles move when considering the surrounding environment (obstacles, pathways, etc.) and the individual's or group's past movement over time. Multi-agent trajectory prediction incorporates social interaction and collective pedestrian or vehicular dynamics by extending ATP to several agents like vehicles, individuals, and autonomous robots.

Such predictions play important role in facilitating autonomous movement and smart surveillance. The main applications extend to:

- Autonomous navigation of vehicles in crowded or pedestrian heavy environment.
- Automatic detection of abnormal behavior by security cameras.
- Navigation of mobile service robots in different locations.
- Modeling and Simulating crowd movement to design practical infrastructures.
- Developing strategies for multiple-player sports, and so on.

Traditional means of ATP involved physics-based and rule-based heuristic models. Development of Markov chains also led to probabilistic modeling but these all fail to consider social interaction and contextual clues. Recent advancements in Artificial Intelligence and Machine Learning has made it possible to simulate trajectories in complex situations. From Recurrent Neural Network (RNN) to Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) to Transformers and Diffusion models, multimodal prediction and contextual integration has been developing rapidly. Use of Graph Neural Networks (GNNs) and Social-LSTM, and Social-Generative Adversarial Network (Social-GAN) has made it possible to capture multi-agent interactions.

However, owing to various reasons like dynamic environment, unpredictable human behavior, and so on, satisfactory prediction has not been possible. Even for sparse environment, prediction relies on heavy computational power. Integrating proper physics-guided interaction rather than letting prediction models learn from inherent agent interaction within the dataset may lead to easier computation while allowing more accurate predictions.

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

1.1 Problem Definition

Despite progress in trajectory forecasting, traditional lightweight trajectory models come with certain limitations such as:

1. Lack the integration of contextual clues and spatial structure
2. Assumption of independence between interacting agents.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar.

1.2 Research Objectives

The main objectives of this study include creating a lightweight vehicle trajectory prediction model that:

1. Predicts the immediate future position of a vehicle by incorporating driving lane information to deliver a more accurate prediction.
2. Integrates physics-guided sparsification to account for dynamic environment with the most important interacting agents.

II. LITERATURE REVIEW

Single agent trajectory prediction has seen a lot of development. Starting from traditional methods like using Kalman Filters [1] to implementing LSTMs to modal temporal dependencies [2] or using GANs to generate multiple future trajectories [3]. For interaction with multi-agent environment, researches such as using Social LSTM for human trajectory prediction in crowded environments [4] to guided multi-graph convolution network for heterogeneous agent trajectory prediction [5] have been realized.

Additionally, some of the recent studies that have explored trajectory prediction and integration with LLMs include, Motion Diffuse [6] that utilizes diffusion models to generate human motion through text prompts for dynamic control, Trajevo [7] that uses LLMs to guide trajectory heuristics through evolutionary optimization, Intention-aware Denoising Diffusion [8] that integrates agent intent into trajectory generation using diffusion techniques and Pedestrian Prediction in Semi-Open Environments [9] that shows how to handle complex interactions in autonomous driving scenarios using diffusion models. Reciprocal Learning Networks [10] that explore mutual influence between agents for social-aware reasoning in multi-agent settings has also been explored.

Few studies on including semantic information in trajectory predictions have also been conducted, such as using context-augmented transformer networks [11], conditioning multimodal outputs on BEV semantic maps [12], or integrating a semantic map with a dynamic graph attention network [13]. When considering integrating both semantic and text descriptions, even fewer works exist. One of the notable ones include multimodal embedding of map visuals and LLM-generated textual descriptions representing trajectory semantics, TrajSceneLLM [14].

Other BEV semantic integrations also include using knowledge graphs for holistic and semantic traffic scene representation [15], probabilistic BEV mapping aggregating LiDAR features and semantic segmentation [16] and context-aware transformer network by integrating CNN with BEV semantic map [17]. These models suffer from the same issue; they integrate extensive features and require heavy computational resources.

When considering lightweight interaction aware models, the paper most followed during this study includes a multi-modal trajectory predictor integrating temporal and spatial interaction with lane awareness [18].

Other works studied include one that highlight sparse interaction modelling in cross-LSTM decoder for trajectory prediction [19] and another that proposes a generic generative neural system with explicit interaction modeling by incorporating relational inductive biases through a kinematics constraint layer with a dynamic graph representation, leveraging both trajectory and scene context information [20].

III. METHODOLOGY

3.1 System Block Diagram

Figure 1 illustrates the general architecture of the proposed trajectory prediction system. The model extracts features from ego trajectory, neighboring agents and driving lane, aggregates them and used different encoding models to process these features as shown in the figure. The output from these lane attention (lane representation), spatial interaction (neighboring agent and interaction representation) and temporal encoding (ego vehicle trajectory representation) modules are fused and decoded to provide the future trajectories.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

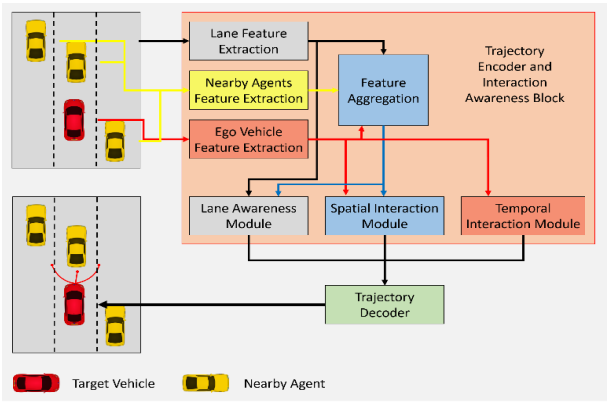


Figure 1: Block Diagram of the Trajectory Prediction System

The following section under chapter 3 explains all these components and workflow of the model in more detail.

3.2 Dataset: NuScenes

The nuScenes dataset, a large scale autonomous driving dataset which is free to use for non-commercial purpose was used for this study.A part of the complete dataset was used to reduce computational overhead and focus on comparison of efficiency and real-time readiness of the proposed model with baseline models. Another smaller part was used to test the models for trajectory prediction and robustness evaluation.

Each sample in the scene contains corresponding annotations that mark surrounding agents through the use of RADAR and LIDAR as shown in Figure 2.

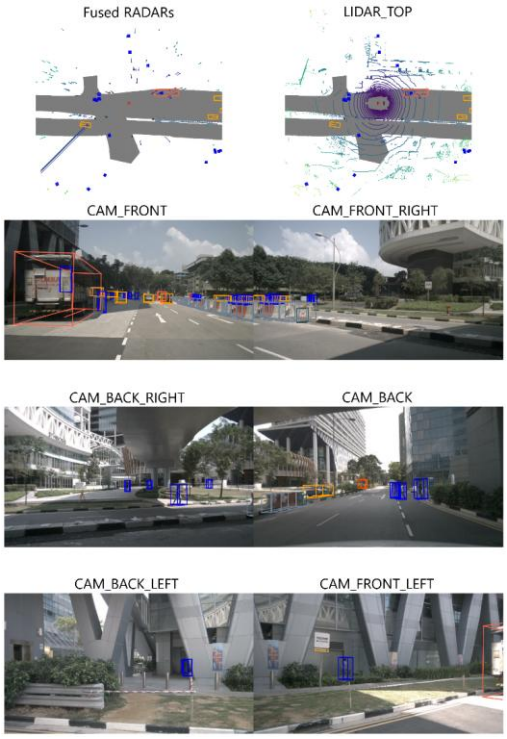


Figure 2: Individual nuScenes sample with visualized annotations

The dataset also includes detailed vector polygons for different layers. As an example, the lane map for one of the map in the nuScenes dataset, Boston Seaport is shown in Figure 3.

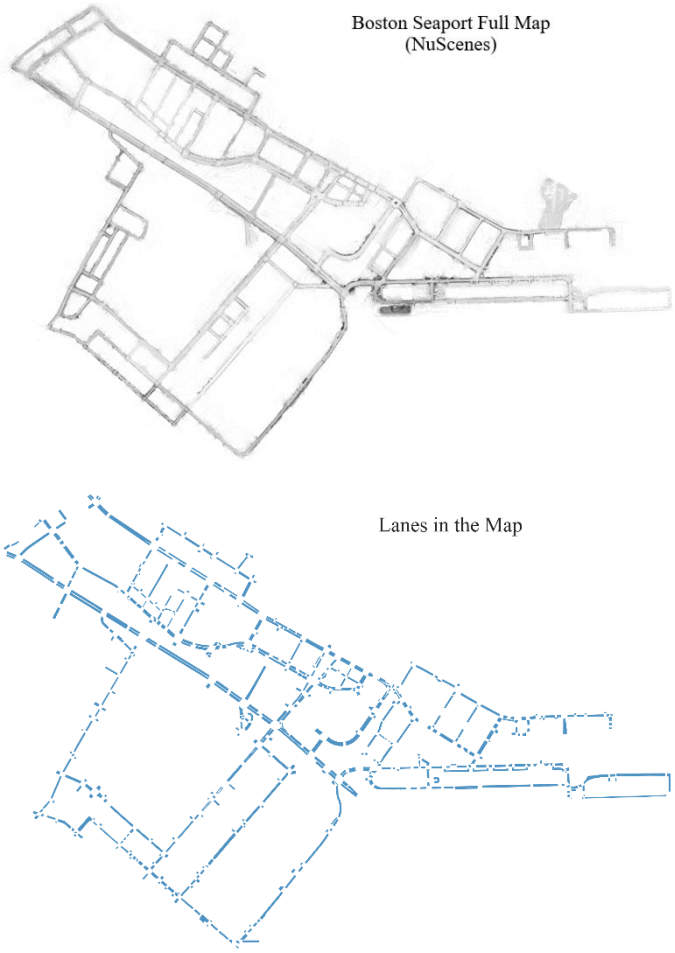


Figure 3: Lane Maps for Boston Seaport Map in nuScenes Dataset

3.3 Data Preprocessing

The raw nuScenes data was transformed into structured tensors for each scene that represent a timestep sequence of N samples, $S = \{s_1, s_2, \ldots s_N\}$.Each scenes is an augmentation of the tensors: **Ego history**, **Ego future**, **Neighbors** and **Lanes**.

a) Ego Agent Trajectories Extraction

Tracked agent states (cars, pedestrians, bikes, etc.) were loaded from the nuScenes dataset, specifically entries in the "sample_annotation.json" file. Each annotation contains positional, orientation, size, velocity and other fields.

For each sample st, the ego vehicle pose from the LiDAR data and agent annotations were extracted as:

$$a_t = (x_t, y_t, q_t)$$

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

where, (x_t ,y_t) represent global position and q_t represents quaternion rotation at timestep t.These values were used to determine velocity v_t, acceleration a_t and yaw angle θ:

$$v_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}$$

$$a_t = v_t - v_{t-1}$$

$$\theta = \tan^{-1}\left(\frac{2(wz + xy)}{1 - 2(y^2 + z^2)}\right)$$

where, (w, x, y, z) = quaternion rotation (q_t) unit vectors

Each trajectory was formed as a timed ordered sequence of T = 6 observed history and H = 12 future predictions, where each timestep is a feature vector:

$$f_t = (x_t, y_t, v_t, a_t, \theta_t)$$

The 6 and 12 timesteps for the observed history and future trajectory were used to cover reasonable short-term prediction. With data sampled at 2Hz or approximately every 0.5 seconds, these correspond with 3 seconds of observation and 6 seconds of prediction respectively.

b) Neighboring Agent Trajectories Extraction

The neighboring agent trajectories follow the same vector representation as the ego vehicle. However, the positions were transformed relative to the ego-vehicle for translational invariance.

$$x'_t = x_t - x_{ref},\ y'_t = y_t - y_{ref}$$

where, (x_{ref}, y_{ref}) is the ego position at time t.

For the model with hard sparsification, top-K (K = 5) agents were selected based on spatial proximity and Time to Collision (TTC), where,

$$TTC = \frac{\sqrt{(x'_t)^2 + (y'_t)^2}}{v_t + \epsilon}$$

and, ϵ is a very small number to prevent the infinity error.

c) Lane Map Representation

The lane centerlines were extracted using the discontinuous lane vector polygons and then resampled to fixed length L = 20 lane polyline points using linear interpolation. Each trajectory uses N = 6 lane candidates. Figure 4 illustrates the entire process.

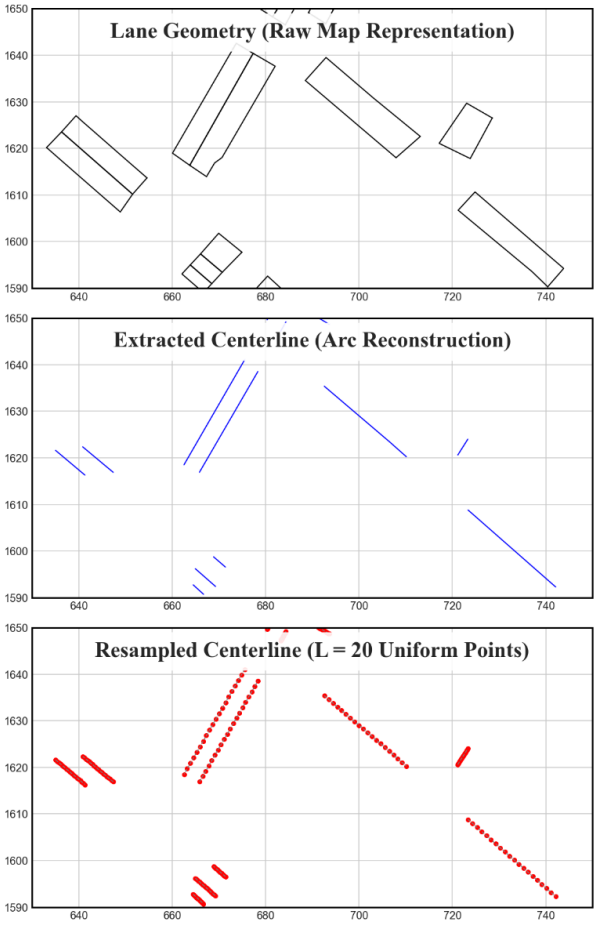


Figure 4: Reconstructing and Resampling Lane Centerlines

3.4 Training and Evaluation

Multiple models with different architecture were trained in this study using PyTorch framework for comparative analysis. All models were regression models trained using supervised learning framework that predict future trajectory timesteps given the past trajectory, lane geometry and neighboring agents.

a) Input Representation

The final processed dataset for each sample contains the following structure and dimensions:

- Ego history: $X_{1:T}^{ego} \in \mathbb{R}^{Tx6}$
- Ego future: $Y_{1:H} \in \mathbb{R}^{Tx6}$
- Neighbors: $X_{1:T}^{nbr} \in \mathbb{R}^{KxTx6}$
- Lanes: $X^{lane} \in \mathbb{R}^{NxLx6}$

b) Feature Encoding

Each component uses LSTM encoders, to retain timed dependencies without requiring heavier computation like transformers and diffusion schemes.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

$$h_{ego}, h_{nbr}, h_{lane} = f_{LSTM}(\cdot)$$

LSTM [21] is an enhancement over RNN [22] and is designed to capture long term dependencies that traditional RNNs fail at. LSTM uses memory cells to store and update important information over time for this purpose.

LSTM encoder-decoder architecture has also proven effective in modeling sequential information with high computational efficiency. All trajectory inputs like change in global position, velocity, acceleration and orientation, lane changes, and so on evolve gradually and require processing the sequences in a step-by-step manner. Traditional MLPs cannot properly maintain hidden states preserving such temporal context and even RNNs are not as suitable for this purpose. A GRU [23] would be a reasonable alternative as it is lighter and less sophisticated, but LSTMs maintain better hidden state retention and are better for the more complex trajectory motion evolution. The

Advanced models like Transformers [24] and diffusion probabilistic models [25] require much in terms on volume of data samples and hardware resources. For a lightweight trajectory model designed for real-time applications, LSTMs provide better generalization with moderate-sized trajectory datasets with lower computational overhead. Since the objective of this study is to explore the feasibility of physics-based sparsification in interaction modeling, using such sophisticated techniques would require accounting for additional architectural complexities.

c) Interaction Modeling

Interaction modeling is the most vital part of trajectory prediction as realistic trajectories do not merely consider the ego agent's past trajectories, but a deeper dynamics of how it handles various interacting agents in its vicinity [4], like pedestrians, other vehicles, traffic lights, etc., as well. Even for lightweight trajectory models, interaction modeling helps keep the prediction realistic, socially compliant and handles potential collision risks.

In this study, a learned score component is used to implement physics in the interaction between ego and neighboring agent using a multi-layer perceptron (MLP) network. An MLP can better capture non-linear dependencies that arises in a complex and dynamic traffic scene.

$$s_i = \mathrm{MLP}\left(\left[h_{ego}, h_{nbr,i}, \phi_i\right]\right)$$

where, ϕ_i (Physics feature vector) $= [d_i, \Delta v_i, \Delta\psi_i]$

where, d_i (Relative distance) $= \| (x_i, y_i) - (x_{ego}, y_{ego}) \|^2$

Δv_i (Relative velocity difference) $= |v_i - v_{ego}|$

$\Delta\psi_i$ (Relative orientation difference) $= |\psi_i - \psi_{ego}|$

Implementing learned interaction scores helps identify the importance of each agents and how they affect the ego trajectory in a better manner than just using some heuristic rules. A combination of physics and learned features adds inductive bias to the latent representation, improving interpretability and learning efficiency. Pure latent representation may not be able to learn various relationships like collision probability, relevance of distance and velocity, and so on.

The specific physics features, relative distance, relative velocity difference, and relative orientation difference were chosen due to their meaningful significance to object interactions and computational simplicity. These three metrics correspond with spatial relevance, dynamic interaction intensity and interaction geometry respectively.

After scoring, sparsification is applied using gated aggregation:

$$\alpha_i = \frac{\sigma(s_i).\,1(i \in s_i)}{\sum_j \sigma(s_j)}$$

$$h_{nbr}^{agg} = \sum_i \alpha_i \,.\, h_{nbr,i}$$

where, σ is the sigmoid function α represents gated weights, normalized to produce stable gradients and consistent scale.

Sparisfication aids in removing irrelevant agents that do not provide meaningful interactions and makes the model more noise resistant. In a realistic traffic environment, only a select few neighboring agents have an influence on the ego trajectory [26]. It also reduces computational overhead, which is one of the goals of this study.

Sigmoid function was applied to bind the potentially unstable scores to interaction probability. Other activation functions used in sparsification, like Softmax, were not considered as they may lead to exclusive interaction where the score for one neighboring agent suppresses others. Realistic motion dynamics may have multiple critical agents and require independent estimation of relevance. Gated aggregation helps model context-aware interaction using learned importance of neighboring agents.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

d) Lane Attention

Vehicle trajectories are heavily constrained by the road topology, so the lane structure has a deep importance on ego agent dynamics. To integrate lane information in the model, the lane polylines were scored using scalar weights representing importance and then softmax functions are applied to convert the score into probability, before being aggregated. Lane attention is global and fully differentiable because only 6 lanes are considered during input representation.

$$w_i = \text{softmax}(W.h_{lane,i})$$

$$h_{lane}^{agg} = \sum_i w_i . h_{lane,i}$$

The lane attention module acts as a soft path selector, where the model can implicitly infer the potential driving area among multiple surrounding lanes. This representation should help align trajectory prediction with the map topology and reduce off-road predictions.

Using encoded lane features $h_{lane,i}$ for this representation is supposed to preserve geometric, directional, and other semantic map features instead of raw lane polylines. The Softmax function helps determine the most critical lane while suppressing all others, reducing noise and ambiguity. This is important as the scene may contain unreachable lanes, intersections and multi-lane drivable area, especially with opposite lane directions.

e) Temporal Ego Encoding

The model also adds global temporal aggregation to capture motion trend over time. The temporal aggregation (h_{ego}^{temp}) is fused with the current state of the vehicle (h_{ego}) so that the model becomes sensitive to both trajectory evolution and instantaneous state.

$$h_{ego}^{temp} = \frac{1}{T}\sum_{t=1}^{T} h_{ego,t}$$

$$h_{ego}^{enh} = h_{ego} + h_{ego}^{temp}$$

Simply using the final state of the ego agent may not preserve the temporal evolution of a sequential ego motion. Especially for decisions like change in lanes, motion over curved lanes, and so on, the instantaneous state fails to represent the intent of the motion. Integrating average pooling over time helps capture the information from the entire sequence and smoothens the transient fluctuations.

f) Fusion and Decoding

All three components are aggregated as a fused vector and then raw ego representation is added to all components to preserve the original trajectory information and prevent significant domination by neighbor and lane features.

$$z = [h_{ego}^{enh}, h_{nbr}^{agg}, h_{lane}^{agg}]$$

$$z = z + [h_{ego}, h_{ego}, h_{ego}]$$

Then an LSTM decoder is used to generate trajectory predictions.

$$\hat{Y}_{1.H} = f_{dec}(z)$$

Since the future trajectories are also sequential in nature, LSTM decoder architecture was naturally chosen to correspond with the LSTM encoder.

g) Training Configuration and Optimization

Since long term dependency neural architectures like LSTM are sensitive to noisy inputs, unstable gradients, inconsistent scaling, and so on, proper architecture tuning and several optimization techniques were used for the training. These include the following:

i) Optimizer

An optimizer is the backpropagation algorithm used in a neural network to update the weights of each nodes in order to reduce the loss function. In all the models created during this study, Adam Optimizer was used with a learning rate of 1x10-3.

Adam Optimizer is one of the more popular optimizers that maintains moving averages of both gradients (first moment) and squared gradients (second moment) for better training stability and faster convergence. This optimizer was selected as it is known to be extremely suited for sequential dependencies with non-linear motion patterns.

Learning rate controls the amount by which the weights in the model are updated during each backpropagation. A small learning rate considerably slows down training while a larger learning rate may cause unstable convergence. The 1x10-3 was chosen as the most stable learning rate after repeated tests for the proposed model along with a few other

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

models for comparison. This value was used for all the models created during this study for fair comparison.

ii) Batch Size and Epochs

The training was performed with batches of 32 samples. While the maximum epoch of 200 was set, an early stopping strategy was implemented with a patience threshold of 15 knowing full well that the maximum epoch limit would never be reached after multiple iterations.

The batch size determines how many samples are processed during one training cycle, i.e., before performing optimization and modifying the weights. Smaller batches improves generalization and are memory efficient, but may introduce noisier gradients and impose a higher load on the processor. On the other hand, larger batches are computationally efficient but show reduced generalization. A batch size of 32 was chosen to strike a balance between these aspects given the size of the training dataset.

The epoch size also directly affects training performance. Using fewer epochs than necessary means poor generalization while higher epochs lead to overfitting. Overfitting means the model is trained more than necessary leading to poor generalization even with high training accuracy. This means that it can fit even the outliers and noise in the training dataset, but deploying the model elsewhere leads to unreliable predictions.

Using an early stopping strategy makes sure that the model doesn't overfit while also reducing the chances of underfitting. A patience threshold of 15 was implemented for the early stopping strategy, i.e., if the error or loss failed to improve after 15 successive iterations, the training was stopped and the weights and biases for the best fit were chosen for the final model.

iii) Gradient Clipping

Gradient clipping is the technique that imposes a maximum limit on the magnitude of gradients to prevent exploding gradients problem. Exploding gradients is the issue of the model weights being excessively large causing the model to be unstable and even producing NaN (Not a Number) values as output.

Gradient clipping prevents this issue and maintains numerical stability. While LSTMs are more robust against exploding gradients compared to RNNs, they are still quite vulnerable to this issue through long temporal sequences, making it ideal to implement this strategy.

In this study, gradient clipping with a clipping threshold of 5.0 was employed to prevent exploding gradients and improve the recurrent sequence learning behavior.

iv) Neighbor Masking

Masking is simply the practice of handling unexpected (usually zero or missing) values during training.

In the nuScenes dataset, some sequences contained fewer than 5 neighboring agents. So zero-padding was used for missing neighbors as some of the models expected at least K=5 samples. However, the zero values would lead to the model learning invalid interactions and cause instability.

As such, masking was used to determine the valid neighboring agents and suppress the padded ones.

$$M_i = \begin{cases} 1 & if neighbor exists \\ 0 & otherwise \end{cases}$$

The mask was later multiplied to final calculated values like attention scores, weights for gating, and so on.

v) Global Normalization

Initially, per-sample normalization was used using the local statistics. However, this led to inconsistencies with the output due to difference in scaling between different sequences. For instance, slow and fast vehicles could have similar trajectories after inconsistent normalization.

To address this issue, global normalization statistics (mean and standard deviation) were calculated for the whole dataset and were used for all sequences for consistent scaling and better convergence stability.

vi) Stationary Scene Filtering

Training the whole of the selected dataset led to minor issues due to the presence of stationary agents. The earlier models lacked any differentiation between ego agents that are in motion and stationary. Consequently, they fails to properly predict any feasible future directions for stationary agents as shown in Figure 5. Furthermore, the prediction for mobile agents may also have been negatively influenced by the stationary agents.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

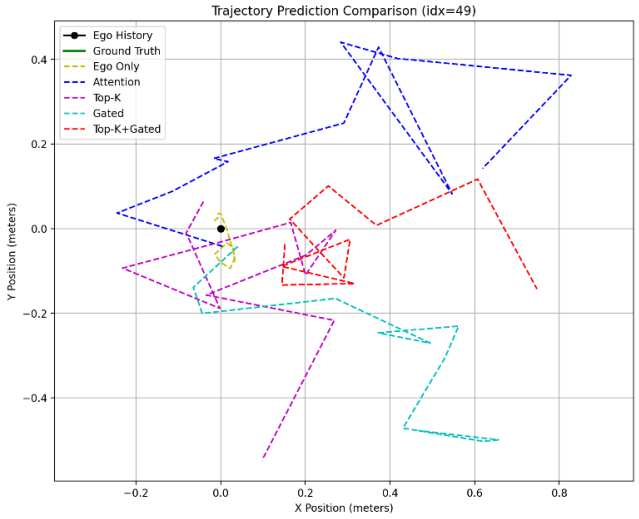


Figure 5: Initially predicted future trajectories for stationary ego agents

To fix the unnecessary bias, the trajectory dynamics of stationary and nearly stationary agents were suppressed by applying a threshold value for the difference between the initial and final position of the ego agent in the sequence. This led to more stable trajectories and better curve prediction.

vii) Selective Feature Normalization

Normalization was only performed on velocity and acceleration information while positional, orientation and lane feature were left untouched. This was done to stabilize motion dynamics without adversely affecting other features. Performing normalization on other metrics would lead to:

- Unstable trajectory shapes as the positional trajectories have already undergone ego-centric conversion.
- Degradation of lane geometry features relative to one another.
- Distortion of the geometry or spatial relationships between interacting agents.

h) Loss Function

A combination of Average Displacement Error (ADE) and Final Displacement Error (FDE) was used to describe the loss function.

- ADE: Average Euclidean distance between predicted trajectory points and the ground truth across all timesteps.
- FDE: Average Euclidean distance between predicted trajectory points and the ground truth at the final predicted point.

Using both metrics takes into account both the average and the accumulated error in the predictor. In short,

$$\mathrm{L} = \mathrm{mean}(\mathrm{ADE} + \mathrm{FDE})$$

where,

$$ADE = \frac{1}{T}\sum_{t=1}^{T} ||\hat{y_t} - y_t||_2$$

$$FDE = ||\hat{y_t} - y_t||_2$$

ADE evaluates the overall consistency of the trajectory prediction throughout the prediction horizon. Lower ADE indicates more accurate trajectory tracking. On the other hand, FDE focuses on the endpoints only. Lower FDE indicates more precise destination forecasting.

For more accurate trajectory prediction, both metrics need to be as low as possible. Lower ADE and higher FDE indicates close prediction during the initial timetseps but significant divergence later. Similarly, high ADE and low FDE indicates unrealistic intermediate motion even if the model predicted the correct endpoint.

i) Other Evaluation Metrics

Apart from the evaluation metrics used in the loss function, other metrics were also calculated for the models during training. These include

i) Gradient Norm

Gradient norm is the scalar measure of the gradient vector and evaluates the magnitude and stability of updated parameters during optimization. Since the gradient is a vector function for all the nodes, L2 gradient norm was calculated for each epoch to monitor the training process. The L2 gradient norm is calculated as:

$$\|g\| = \sqrt{\sum_i {g_i}^2}$$

Where, g_i represents the gradient of parameter i in the gradient vector.Very low gradient norm indicates weak convergence or vanishing gradients while high values indicate unstable convergence or exploding gradients.

This metric was calculated to check for training stability while also verifying the effectiveness of gradient clipping.

ii) Smoothness

Smoothness for trajectory sequences measures the temporal continuity of the predicted trajectories. A second order finite difference acceleration penalty was used to measure the smoothness of the predicted motion.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

$$a_t = \hat{p}_{t+1} - 2\hat{p}_t + \hat{p}_{t-1}$$

where, $\hat{p}_{t-1}, \hat{p}_t, and \hat{p}_{t+1}$ are sequential predicted positions and at represents estimated acceleration at time t. Ideally, a_t should be 0 but for practical consideration, lower value implies smooth turning (for curved path) or nearly constant motion (for straight path). Higher value indicate sudden and unrealistic changes in trajectory direction.

Finally, an average of all the acceleration metrics over the predicted horizon was calculated as the smoothness score.

$$Smoothness = \frac{1}{H-2}\sum_{t=2}^{H-1}\|a_t\|$$

where, H is the prediction horizon.

Since Smoothness averages the acceleration metrics, its physical significance is the same as the significance for an individual acceleration penalty.

iii) Collision Rate

Collision rate determines the frequency of intersection between predicted trajectories and neighboring agents' positions within a predetermined safety threshold.

When calculating the distance between the predicted trajectory point and the neighboring agents, a distance lower than the threshold registers as a collision.

$$CollisionRate = \frac{No.\ of Colliding Trajectories}{Total No.\ of Trajectories}$$

This metric evaluates safety awareness of the model. A model that produces low ADE and FDE but high collision rate is still not safe for deployment.

iv) Per-Horizon Error

Per-Horizon Error measures independent temporal errors at each future time steps. Since error in trajectory prediction usually accumulates for each timestep, Per-Horizon Error gives insight into both the short-term prediction stability and long-term prediction degradation.

Per-Horizon error can be analyzed as an error curve across the predicted horizon, where the error at timestep t is:

$$E_t = \|\hat{p}_t - p_t\|$$

3.5 Robustness Evaluation

For this study, robustness evaluation was conducted using the a smaller test dataset to analyze the behavior of all the models under various stress conditions resembling the real-world incomplete, degraded or corrupt input data conditions.

Real-time data collected during autonomous driving may contain inconsistencies like incorrect lane information, missing sensor measurements, noisy positional and velocity data, and so on. As such, different robustness tests were conducted to analyze sensitivity of the model to input perturbations and resilience under degraded conditions.

To evaluate performance during such stress testing, four evaluation metrics, namely, ADE, FDE, Smoothness and Collision Rate were used.

a) Lane Removal Test

The Lane Removal Test evaluates the dependency of the predictor models on the lane geometry. All of the lane attention inputs were replaced by zero tensors for the purpose of this test to simulate the extreme condition where all lane representation is suppressed.

$$h_{lane} = 0$$

This test analyzes the nature of degradation of the model and its stability without lane guidance. A more robust model should be able to generate physically plausible motion, albeit with gradual degradation, just based on past trajectories and neighbor interactions.

b) Zero Neighbor Test

The Zero Neighbor Test evaluates the dependency of the predictor models on the surrounding agent interactions. All of the neighbor representations were replaced by zero tensors for the purpose of this test to simulate the sparse traffic condition without any interacting agents.

$$h_{nbr} = 0$$

This test analyzes whether the predictor relies heavily on just social context, along with the safety and stability performance when nearby agent trajectories are missing or corrupt. It also evaluates the effectiveness of the interaction sparsification, neighbor masking, and motion encoding used in the training.

A more robust model should be able to generate physically plausible and safe motion even when social context is missing.

c) Noise Robustness Test

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

The Noise Robustness Test evaluates the sensitivity of the predictor models to noisy positional input parameters. This test simulates realistic autonomous vehicle conditions where sensor data for Global Positioning System (GPS), LiDAR, camera, etc., may show inconsistencies and inaccuracies, leading to temporal tracking instability.

For this test, Gaussian noise was added to the positional inputs during inference of future trajectories.

$$\tilde{p} = p + N(0, \sigma^2)$$

Where,
p = original trajectory position

$\tilde{p}$ = corrupted position after adding noise

$N(0, \sigma^2)$ = Gaussian noise with 0 mean and σ standard deviation

Multiple noise parameters, namely with standard deviations 0.5, 1.0 and 2.0, were added for each input to analyze the response of the models to growing positional noise.

A more robust model should be able to maintain relatively stable motion even without relying excessively on precise spatial positioning.

IV. RESULTS AND DISCUSSION

This section presents the description and capability comparisons of the deployed models, including the proposed model, training performance of each model under various evaluation metrics, and qualitative as well as quantitative analysis of the experiment results from the deployed models.

4.1 Model Architectures and Inference Benchmarking

Five model architectures were used to train the trajectory dataset and the best models for each architecture were selected for the comparative analysis of the proposed model. This section explains the architectures, components used in the architecture and the theoretical computational efficiency of the models based on similar conditions.

The five selected predictor models based on the architecture include:

- **Model 1: Ego-only predictor**: Only uses the vehicle's observed history while ignoring the agents and lane information.
- **Model 2: Attention-based predictor**: Implements attention mechanism over all neighboring agents.
- **Model 3: Top-K predictor**: Hard selects top-k neighboring agents to reduce computation overhead and noise.
- **Model 4: Gated predictor**: Uses gated mechanism for adaptive sparsification while using physics metrics during gating as inductive bias.
- **Model 5: Gated + Top-K predictor (Proposed model):** Combines both hard selection of critical agents and physics based sparsification through adaptive learning for balanced performance.

To better express the critical components (excluding generic ones used in all models) used in each model, an ablation-style comparison was conducted that summarizes the architectural capabilities, as shown in table 1.

Table 1: Comparison of all deployed models based on architectural components

Models / Component	1	2	3	4	5
Ego trajectory encoding	√	√	√	√	√
Neighbor trajectory encoding	×	√	√	√	√
Lane encoding	×	√	√	√	√
Lane attention mechanism	×	mean-based attention	mean pooling	learned attention	learned attention
Soft attention (interaction modeling)	×	√	×	×	×
Explicit Top-K selection	×	×	√	×	√
Learned gating mechanism	×	×	×	√	√
Physics-informed scoring	×	×	×	√	√

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

Neighbor masking	×	√	√	√	√
Lane influence weighting	×	√	√	√	√
Residual fusion	×	√	√	√	√

This components comparison also illustrates the progressive architectural evolution from ego-centric prediction to dense interaction modeling and then to physics-based sparsification for interaction reasoning, which was the core consideration for selecting these particular model architectures.

Since the goal of this study is to facilitate real-time inference for autonomous driving using a lightweight model, all model architectures were compared based on inference benchmarking to analyze deployment feasibility. The comparison is shown in table 2.

Table 2: Comparison of all deployed models based on architectural components

Model	Inference Benchmarking				
	Parameters	**Mean Batch Time**	**Std Batch Time**	**ms/sample**	**FPS**
Ego Only	226,456	15.09	8.12	0.47	66.28
Attention-based	259,224	19.82	5.46	0.62	50.44
Top-K	259,224	16.44	2.37	0.51	60.82
Gated	292,762	17.04	7.70	0.53	58.68
Top-K + Gated (Proposed)	292,762	16.95	2.34	0.53	59.00

Parameters represent the total number of trainable weights in the neural network. From table 4.2, it can be seen that the Ego-only predictor has the lowest number of parameters due to lack of lane and interaction integration. Parameters for Attention-based predictor increases due to addition of those components, and their corresponding attention mechanism. Top-K predictor does not change the neural network size, so the parameter count remains the same as Attention-based predictor. Gated and Top-K + Gated predictors have the highest parameter count due to added computations and layers for gating.

Mean Batch Time indicates the average time used to process a batch (here, size = 32) during learning. The **ms/sample** shows the inference time for an individual sample while **Frames Per Second (FPS)** is the inverse of Mean Batch Time and indicates the number of batches the model can process in a second. Lower Mean Batch Time, lower ms/sample and higher FPS all indicate faster inference and real-time suitability. From table 4.2, it can be observed that the Ego-only predictor has the least computational overhead, albeit it also shows the least accuracy as discussed later. Attention-based predictor has the least computational efficiency and the other three predictors show somewhat similar performance with the order from most efficient to least being Top-K predictor, Top-K + Gated predictor, and Gated predictor respectively.

From the difference in performance between Attention-based vs. Top-K predictor and that between Gated vs. Top-K + Gated predictor, it can be seen that sparsification (hard top-K selection) reduces computational overhead and improves inference time and performance without having to increase the model size.

Standard Deviation of Batch Inference Time (Std Batch Time) shows the fluctuations between the batches during inference. Lower Std Batch Time indicates stable and consistent inference. Models with high Std Batch Time are not suitable for real-time deployment due to the unpredictability of the worst case latency and risk of failing to meet real-time deadlines. From table 4.2, it can be observed that Ego-only and Gated predictors have comparatively higher Std Batch Time and are less suited for real-time deployment, especially compared to Top-K and Top-K + Gated predictors that show the least deviation.

Purely from observing the inference benchmarking of all these models, Top-K and Top-K + Gated predictors are the most suitable for real-time driving applications due to their efficient inference calculations and better temporal reliability.

If the Top-K + Gated predictor shows better generalization in practice compared to Top-K only, as it theoretically should due to the architectural complexity and higher parameter count, it would be the most efficient motion prediction model among those compared in this study as the inference metrics for these two models show little difference.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

4.2 Training Dynamics and Evaluation Metrics

This section focuses on quantitative analysis of the convergence characteristics of each model, including loss reduction trends, optimization stability, inherent safety considerations, and degree of error accumulation across the entire prediction horizon.

a) Training and Validation Loss Convergence

The training performance of these models can be observed through the entire trend of loss stabilization in figure 6 and the final evaluation metrics shown in table 3.

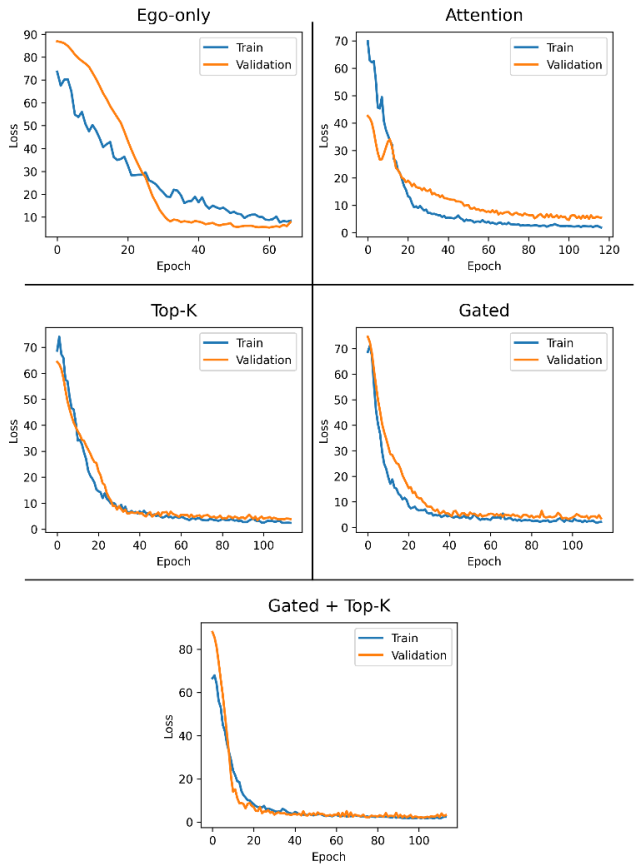


Figure 6: Training vs. Validation losses for all five trajectory predictor models

The training vs. validation convergence plots (figure 6) illustrates the learning behavior and generalization capacity of all the models.

All models exhibit gradual decrease in the trajectory loss with increasing epochs, indicating successful feature learning given the architectural considerations. However, significant discrepancies can be observed between the plots regarding slope characteristics and convergence rate, which directly highlights the effectiveness of the architectural components used for the models.

The Ego-only model shows the slowest convergence, indicating lack of proper generalization of the motion predictor without vital lane geometry and social clues. Initially the validation loss is very high but rapidly converging to a minimum loss range within 30 epochs before plateauing. The training loss curve is almost linear throughout the training, which shows that the model stopped generalizing on the validation set even without learning all latent information about the training dataset. The validation curve also crosses the training loss curve at around 25 epochs, after which the validation loss remains lower than the training loss. This behavior that is contrary to general training convention also highlights the model's instability. It also indicates that faster generalization due to lack of all critical parameters does not mean more accurate optimization.

For the Attention-based predictor, the initial low validation error indicates slightly better initial generalization as the model could successfully capture coarse relations between the interacting agents. However, this curve also shows the lowest slope, indicating that allocating equal importance to all parameters (neighboring agents) also adds noise due to less relevant parameters and pose a detrimental effect on the convergence and optimization.The Top-K Predictor and the Gated Predictor both show similar training and validation curves with faster convergence and better slope optimization compared to Attention-based Predictor, indicating either sparsification or gating provides a positive influence on the learning process.

The proposed hybrid Top-K + Gated model however, had the best convergence rate and highest slope until the low-improvement convergence region, indicating that combining both Top-K selection to select the most critical agents and applying learned importance with physics-based inductive biasing showed the best learning behavior.

Table 3: Comparison of final training performance for all predictors

Models	Epoch	Training Losses			Validation Losses		
		ADE	FDE	Total	ADE	FDE	Total
1	61	2.96	5.79	8.76	1.85	3.58	5.43
2	101	1.17	1.23	2.41	1.74	2.89	4.63
3	98	1.34	1.72	3.06	1.42	1.99	3.42
4	104	1.14	1.62	2.77	1.18	2.07	3.25
5	98	0.91	0.91	1.83	0.77	1.13	1.91

Table 3 expands upon the conclusions drawn from figure 6. Here, the Ego-only predictor converged the quickest, but with the worst training and validation accuracy. This further proves that it failed to draw enough meaningful inferences from the dataset even before the model began overfitting. Attention-

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

based predictor shows reasonable training accuracy, but the validation losses are too high, likely due to similar reasons as stated before. Top-K and Gated Predictors show similar results with the Gated Predictor being slightly better in comparison, albeit with slower convergence.

The proposed hybrid model, however, converged similarly to the Top-K predictor but with much better training and validation accuracies. The ADE loss in the validation dataset are even slightly lower than the ADE loss in the training dataset. This is both because the global and feature selective normalization helps in better generalization and use of gated learning and hard selection of critical agents helped reduce irrelevant interactions leading to smoother motion trajectories over the prediction horizon.

The validation ADE and FDE curves of all the models were also compared together in this study as shown in figure 7.

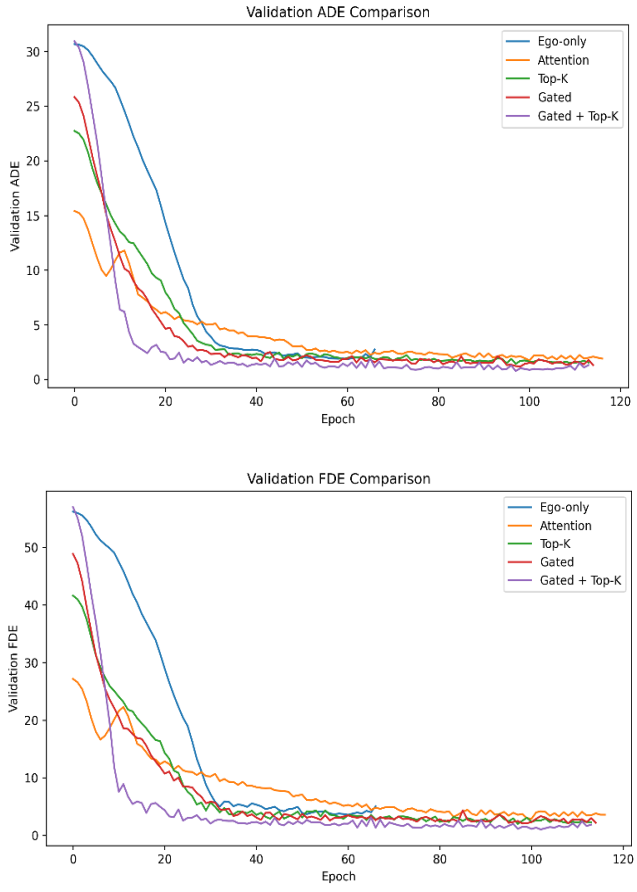


Figure 7: Validation ADE and Validation FDE comparisons between all models

ADE measures overall trajectory consistency and FDE measures long-term prediction accuracy. Since the plots for both metrics are similar overall, just comparing the validation loss plots that combine both metrics would give similar inferences. So most of the conclusions that can be drawn from these graphs have already been discussed. The main inference that can be made is that Ego-only and Attention-based predictors are the worst in comparison at both predicting overall trajectory and predicting long term trajectories, while the proposed hybrid model excels at both.

b) Optimization Stability Analysis

Two evaluation metrics, Gradient Norm and Smoothness were calculated during training. The Gradient Norm was measured for the testing cycles and Smoothness was calculated for the validation cycles in each epoch. The plot of Gradient Norm vs. Epoch can be seen in figure 8.

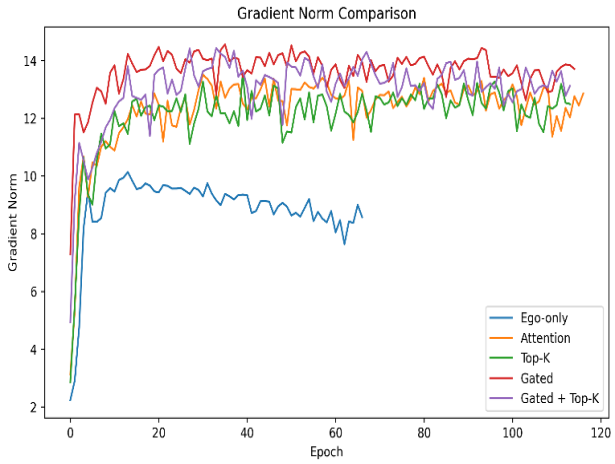


Figure 8: Gradient Norm comparison between all prediction models

As shown in figure 8 and table 4, the Gradient Norm for all models are relatively low, indicating the efficacy of Gradient Clipping used during training. An earlier comparison without implementing this algorithm showed Gradient Norm of values greater than 100 for all models, indicating exploding gradient. The final versions are more stable in comparison.

The Ego-only Predictor shows the least Gradient Norm, and consequently, the most stable parameter optimization. However, this is mostly due to limited optimization dynamics and less-complex architecture.

The Gradient Norm for all other predictors are slightly larger and fall under the same range of values, indicating stable update of optimization parameters. The exact order from overall highest to lowest values (from the graph): **Gated > TopK+Gated > Attention-based > Top-K** correspond to the complexity and size of the model architecture, and more or less similar values when combined with inferences from validation losses show that there were no major issues with stability during the learning process.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

Table 4: Comparison of final stability metrics during training for all predictors

Models	Stability Metrics	
	Gradient Norm	Smoothness
Ego Only	8.0443	0.3423
Attention-based	13.1701	0.478
Top-K	12.891	0.3618
Gated	13.758	0.451
Top-K + Gated (Proposed)	13.686	0.3984

The comparison for Smoothness can be observed from table 4 and figure 9. Lower smoothness is always preferred as it shows smooth changes in the trajectory motion, velocity, acceleration and orientation.

Figure 9 shows that initially during the early epochs **(0 – 15)**, the Smoothness for all models were significantly high, with the highest to the lowest order being **Gated > Top-K+Gated > Attention > Top-K > Ego-Only**. This behavior illustrates the initial instability caused by the complexity of the architecture, as this order mirrors the decrease in complexity and size of those models. As the training progressed, the order for middle epochs **(15 – 40)** becomes **Ego-Only > Top-K > Attention > Gated > Top-K+Gated**, indicating that the more essential interaction-aware models learned more consistent trajectory representation, compared to models with lacking or weaker interaction-awareness.

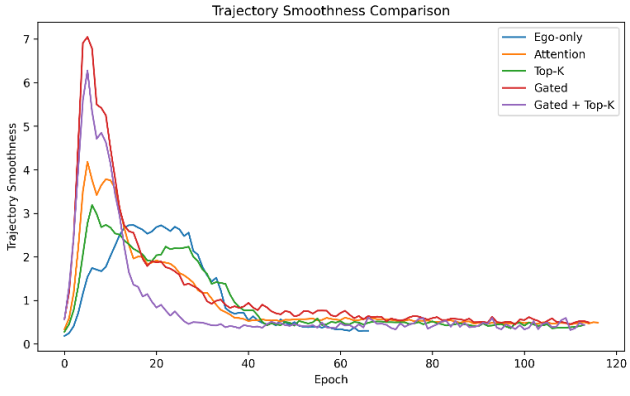


Figure 9: Trajectory Smoothness comparison between all prediction models

After the stabilization of parameters in the later epochs, the order as seen from table 4 is **Attention > Gated > Top-K+Gated > Top-K > Ego-Only**. The least smoothness of the Ego-Only predictor does not directly indicate better prediction quality. When combined with results from other evaluation metrics, it can be said that it is a result of simplified trajectory prediction, and just Smoothness cannot be taken into consideration when determining the most efficient model. The smaller values of Top-K and Top-K + Gated predictors compared to Attention-based and Gated Predictors directly show the positive influence of sparsification on prediction stability.

c) Safety-Oriented Metrics

The evaluation metric, Collision Rate was determined during model training to analyze whether the models could be safely deployed. A trajectory predictor that shows very low ADE and FDE, but high Collision Rate is still useless.

Figure 10 and table 5 both compare the collision rates for all the models. From the figure, it can be seen that the Top-K and Top-K + Gated predictors had very high collision rate initially (< 10 epochs) before decreasing to stable limits while the rest of the predictors exhibited such stable collision behavior from the very beginning.

This is a result of aggressive sparsification through hard selection of K=5 interacting agents that led to sharper and more interaction-sensitive predictions during the early optimization phase. After around 20 epochs, the collision rates for all models fall under 0.0 and 0.1 indicating more of less safe behavior.

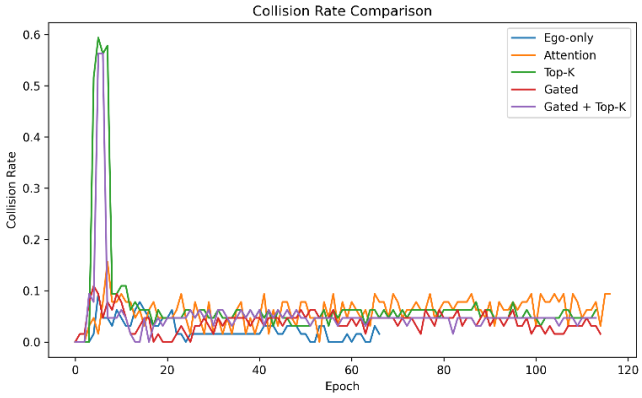


Figure 10: Collision Rate comparison between all prediction models

To analyze further, table 5 that summarizes the final collision rates for the selected models can be observed. According to the table, the Ego-Only Predictor had zero collision rate, theoretically indicating safest behavior. However, this value is misleading. By also analyzing the accuracy of this model, it is more likely that this model predicts trajectories on non-lane areas due to its simplified prediction.

The Top-K, Gated, and Top-K + Gated all show the same values, but this is a lucky happenstance more than anything as the plot all show variations throughout for these models.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

While their collision rates are slightly higher than that of the Attenion-based predictor, it is not that high, and reflects the architectural design to focus more on importance of each agent or just the critical agents rather than on all interacting agents in order to improve accuracy of prediction.

Table 5: Comparison of final safety metrics during training for all predictors

Models	Safety Metrics: Collision Rate
Ego Only	0
Attention-based	0.0312
Top-K	0.0469
Gated	0.0469
Top-K + Gated (Proposed)	0.0469

Finally, all these Collision Rate values, when considered in isolation, are low enough that the models should be safe for deployment. Any amount of trajectory prediction training can never reduce the Collision Rate to zero without causing overfitting or creating significantly low accuracy. In a realistic scenario, the neighboring agents will also manually change their orientation and acceleration based on the trajectory changes of the ego vehicle to prevent collision.

d) Per-Horizon Prediction Error Analysis

Per-Horizon Prediction Error shows the temporal degradation of motion prediction with each timestep. Figure 11 shows the Prediction error of all the prediction models through the 12 timesteps of the prediction horizon.

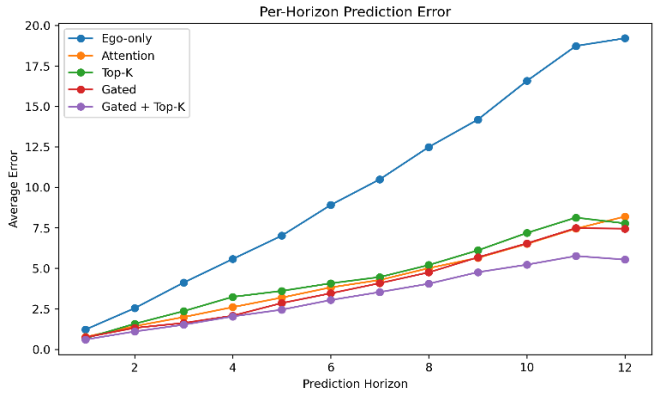


Figure 11: Per-Horizon Prediction Error plot for all predictors

According to the figure, the Ego-Only baseline model shows the highest prediction errors across all timesteps while the proposed hybrid (Top-K + Gated) model shows the lowest. Other models shows errors that are much closer to the error values of the proposed model, but still slightly higher. From the figure, it can be concluded that the long-term trajectory drift accumulates rapidly if the model does not integrate any spatial and social clues. Integration of these inputs in any manner will significantly reduce the long-term error accumulation, but even among all the other models, the proposed model had the best representation of interaction context across the prediction horizons.

4.3 Qualitative Trajectory Prediction Analysis

In addition to the quantitative analysis using various evaluation metrics and robustness tests, qualitative analysis through actual trajectory prediction from the created models were also conducted to reach a deeper insight into the behavior of each model.This analysis follows visual inspection to analyze the trajectory smoothness, stopping/starting responses, turning behavior, directional consistencies, and so on for different motion conditions.

a) Interpretation of Trajectory Visualization Results

The trajectory prediction curves used for this report follows the same composition as figure 12. Here,

- Each plot follows a sequence of 6 historical trajectories as input and 12 future trajectories as output.
- The X and Y axes show relative position in meters for a top-down view of the ego vehicle centered around the last position on the observed history.
- The black line with black circles (representing timesteps) represents the historical trajectory of the ego vehicle used as input in the sequence.
- The green line represents the actual future trajectory of the ego vehicle used to compare the quality of predicted future trajectories.
- The dotted lines represent the predicted future trajectories for each model and is color coded as shown in the legend.

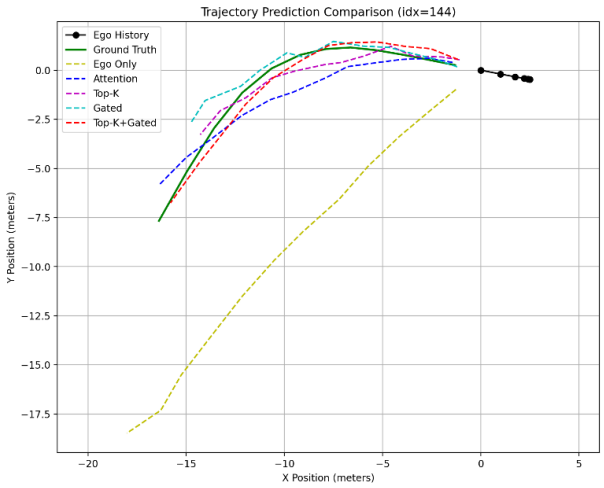


Figure 12: Trajectory prediction comparison for continuation of starting motion of the ego vehicle into a curvilinear trajectory

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

When interpreting the plot, the most important characteristics to be considered include:

- **Trajectory Alignment**: The degree of alignment of the predicted trajectory with the actual ground truth directly determines the spatial accuracy of the model. The consistency of the intermediate paths corresponds with the ADE and evaluates the overall prediction capability. Lack of alignment in direction and shape (e.g., curvature) indicates instability and incorrect motion reasoning. Curvature preservation should also be considered well as it evaluates the direction and lane geometry representation of the model.
- **Endpoint Deviation**: The endpoint of the predicted and ground truth plots represent the 12th timestep for the future trajectory. The distance between these endpoints corresponds with the FDE and evaluates the long-term prediction capability and error accumulation of the model. Since velocity and acceleration were also included in addition to spatial position as trajectory input, high endpoint deviation indicates the model failed to generalize these parameters properly.
- **Physical Plausibility**: Realistic trajectories are smooth, and shifts in direction is more gradual. Jagged, oscillatory or random trajectories indicate abrupt motion changes and are not physically plausible.

Figure 12 illustrates a sequence of continuation of starting motion of the ego vehicle into a curvilinear path. The starting motion can be recognized due to overlap of the timesteps (circles) of the ego history line. In the figure, the ground truth, which is the actual prediction, shows a smooth curvilinear path.

For the predicted trajectories, the prediction of the Ego-only Predictor shows significant deviation in both trajectory alignment and endpoint deviation, although the final direction of the predicted path is more or less similar to the ground truth. This highlights the lack of capability of the model to generalize motion for starting motion and curved paths.

The trajectories for other models follow the curved ground truth but are more or less jagged and show varying amounts of endpoint deviation. Among those, the Attention-based and the Top-K predictor show the least alignment with the ground truth, which shows poor directional and positional reasoning. The Gated predictor showed good alignment before encountering significant change in direction after which it deviated completely and had the highest endpoint deviation among the four non-Ego-only model. This is likely a result of inclusion of irrelevant neighboring agents while changing direction. The proposed model (Top-K + Gated), however, shows the best alignment and the least endpoint deviation, proving its effectiveness in both continuing motion from rest (i.e., representation of acceleration), curvilinear motion performance, lane awareness and critical interaction context reasoning.

b) Performance in Linear Motion Scenarios

Linear motion analysis evaluates the ability of the predictor model to maintain straight trajectories in a stable manner.

Figures 13 and 14 shows two trajectory sequence plots of linear motion prediction from two different direction, where both the input observed history and the output ground truth represent a constant motion through a straight path. The figure shows that the predictions from all models had more or less good alignment with the ground truth line with low endpoint deviation. This result is understandable as it to extremely easy to generalize such motion, especially since the motion does not require changing lanes or avoiding interacting agents. Even without any trajectory prediction, a simple linear interpolation model would also provide a similar result in this case, so other trajectories still need to be observed for proper qualitative analysis. Despite the similarity in direction and spatial positioning, one of the plots show that the proposed Top-K + Gated model still had the best alignment and the least endpoint deviation.

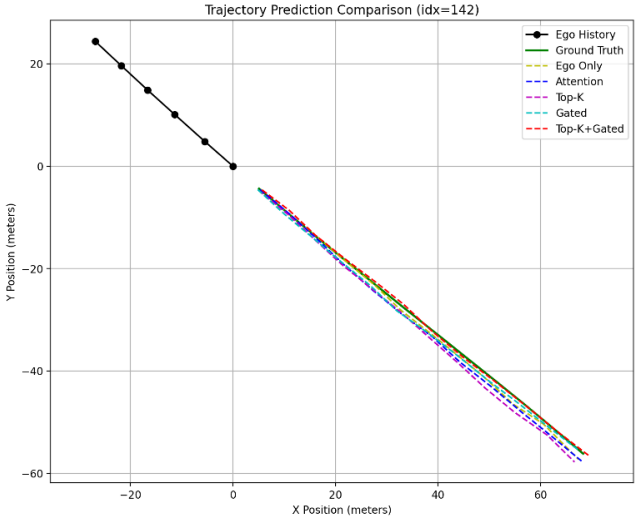


Figure 13: Linear motion future trajectory prediction comparison considering linear motion input for observed history (sample 1)

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

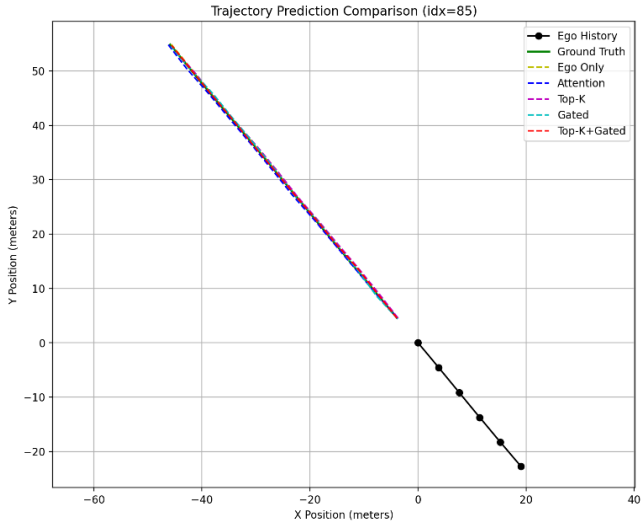


Figure 14: Linear motion future trajectory prediction comparison considering linear motion input for observed history (sample 2)

Apart from just continuation along a linear path, it is important to analyze transition from a curvilinear trajectory path to a straight one as well, as shown in figure 15. Such transition is represents change in lane geometry or impact of neighboring agents.

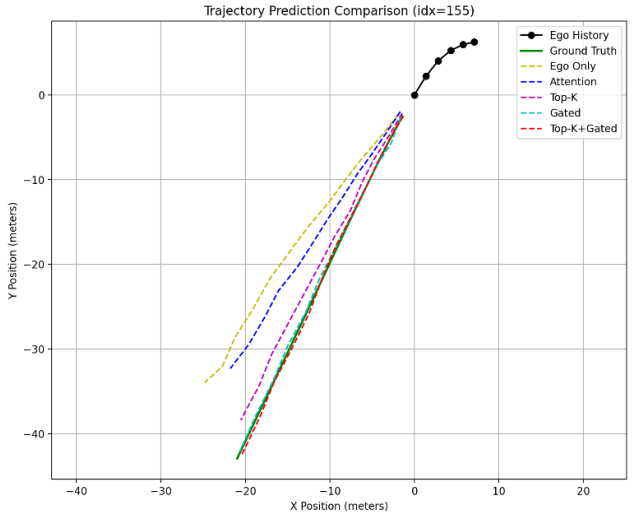


Figure 15: Linear motion future trajectory prediction comparison considering curvilinear motion input for observed history

In the figure, only the Gated Predictor and the Top-K + Gated Predictor showed close alignment to the ground truth path. The other models, namely Top-K, Attention-based and Ego-Only show increasing deviation in the respective order. This shows that Top-K + Gated and Gated Predictors have the best generalization for straight trajectory predictions regardless of the nature of the input (linear or curvilinear).

c) Performance in Curvilinear Motion Scenarios

Vehicle turning motions along a curvilinear path requires more complex understanding of lane geometry, direction transitions, and the turning intent may be driven by nearby agents as well, for instance to avoid collisions. As such, the performance in such motion scenarios are the most important qualitative factors to be considered when evaluating a trajectory prediction model.

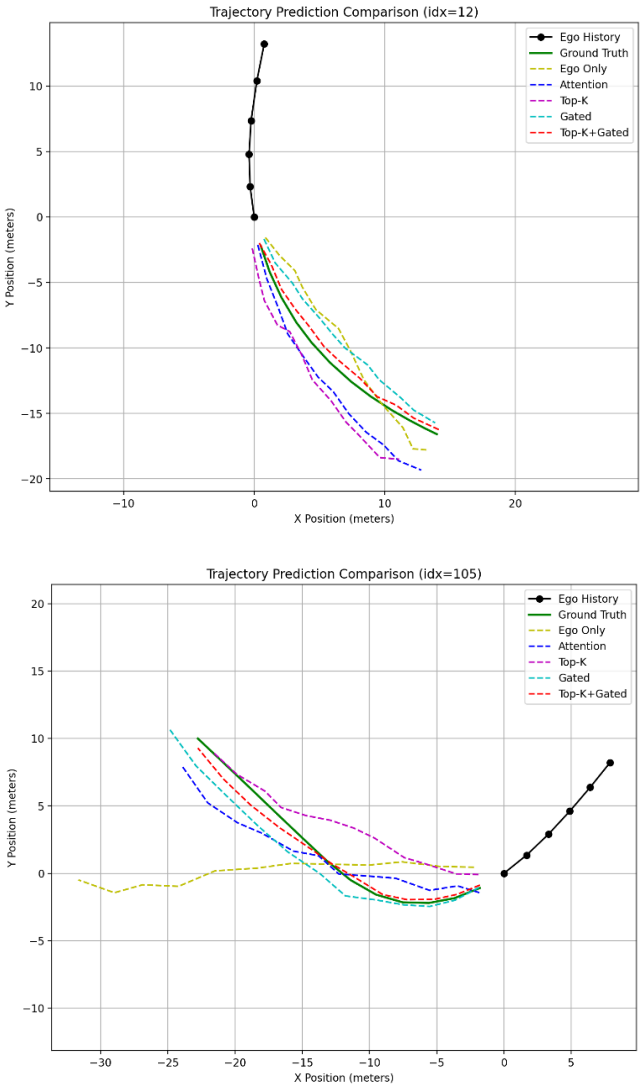


Figure 16: Curvilinear motion trajectory prediction comparison for all predictors

Figure 16 shows two trajectory sequences of the vehicle maintaining a curvilinear motion throughout observed history and the prediction horizon.

According to the figure, the Ego-only Predictor has the worst accuracy and one of the plots even shows a completely different trajectory choice, indicating that lane geometry and social context reasoning are the most important factors for curved trajectory prediction. As with other motions, the Top-K

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

+ Gated model had the best accuracy, followed by the Gated prediction model. Attention and Top-K predictors showed were less accurate with sharp changes in direction instead of gradual turning.

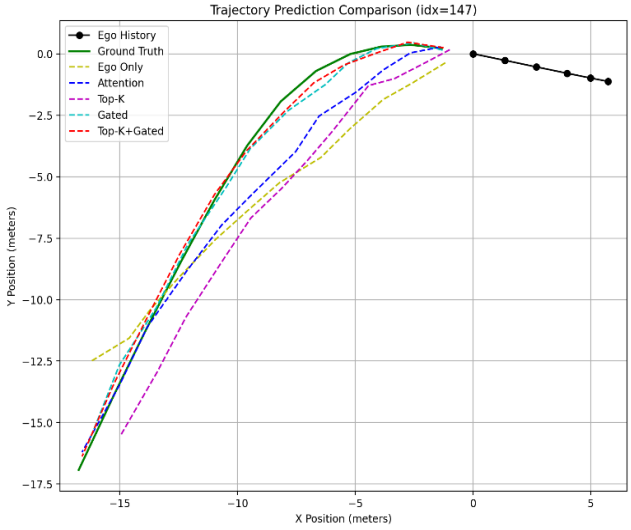


Figure 17: Curvilinear motion future trajectory prediction comparison considering linear motion input for observed history

Figure 17 shows the comparison for motion transition from straight to curvilinear path and the result mirrors the previous one, i.e., trajectories maintaining a curved motion with minor differences. However, in this case, while the Top-K + Gated predictor retains the best alignment and endpoint deviation, the results for the Gated predictor are much closer in comparison with the previous sequence plots, proving that physics-based gated learning provides highly efficient interaction awareness for such lightweight models.

d) Performance during Stopping Maneuvers

Generalization of stopping maneuvers are extremely challenging for any prediction model as it represents a transition from a state of motion to rest, and unless specific algorithms or methods to identify such motion are included during training, the models will fail to recognize such intent. Since most of the data includes vehicles that remain in motion, they will inadvertently affect any small number of samples that represent stopping motion.

Figure 18 perfectly illustrates this issue. The short green ground truth line, shrinking gap between the successive timesteps in the ego history, and the small range of meter values in the axes of this plot, all indicate that this trajectory sequence involves the vehicle coming to rest from a state of motion.

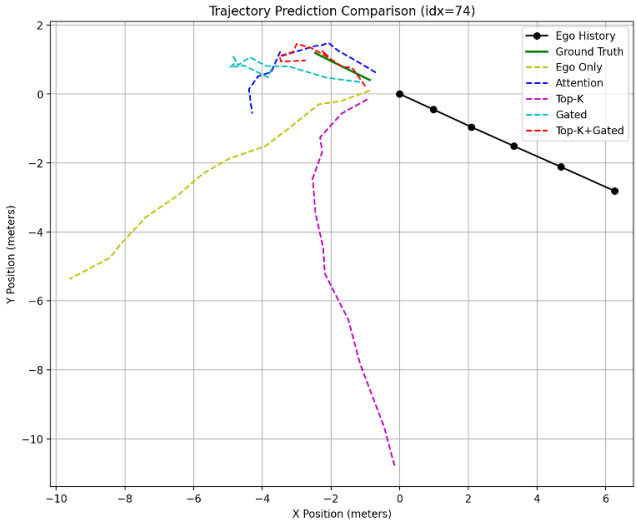


Figure 18: Trajectory prediction comparison during vehicle stopping motion

In the figure, the predicted trajectories show irregular paths and limited alignment with the ground truth trajectory, especially for the Top-K and the Ego-Only predictors. While the other predictors show better conformity to the ground truth, their prediction show a tendency to overshoot before returning to realign with the true motion path using non-realistic trajectories. Such overshoots and irregularity suggest that none of the models were able to generalize the deceleration motion, especially at low velocities.

e) Summarization of the Qualitative Analysis

The inferences from all the plots discussed above helps summarize the effectiveness of each model for different motion scenarios. All figures undoubtedly highlight the proposed model (Top-K + Gated) that combines physics-based learning and sparsification as the most effective one among those compared in the study.

The reason for such high performance of the proposed model can be summarized as follows:

- The integration of physics features helped reduce irrelevant neighbor interactions while biasing the model favorably.
- Real world interactions are sparse and a dense model introduces more noise than necessary. Sparsification also improves computational efficiency.
- The lane attention helped involve only the critical lanes and constrained trajectory prediction in favor of feasible drivable areas.
- Using temporal aggregation through mean pooling instead of direct positional features helped enhance trajectory consistency even in curved areas.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

4.4 Robustness and Generalization Analysis

Three robustness and generalization tests were conducted as stated in the methodology to evaluate the resilience of the model. The analysis from such stress tests helps determines how properly the model can function even in extreme conditions. We can use this analysis along with the quantitative evaluation from training results (under ideal conditions) to evaluate how well the models will perform in realistic scenarios.

a) Lane Context Ablation Study

This study analyzes results from the Lane Removal Test that evaluates the degradation of the models by removing all lane context. In the study, both the absolute tested values of the evaluation metrics (ADE, FDE, Smoothness, Collision Rate), and their degradation from the original baseline is compared. The actual values represents the operational capability of the models under degraded conditions, and the degradation evaluates the dependency of the model on lane context.

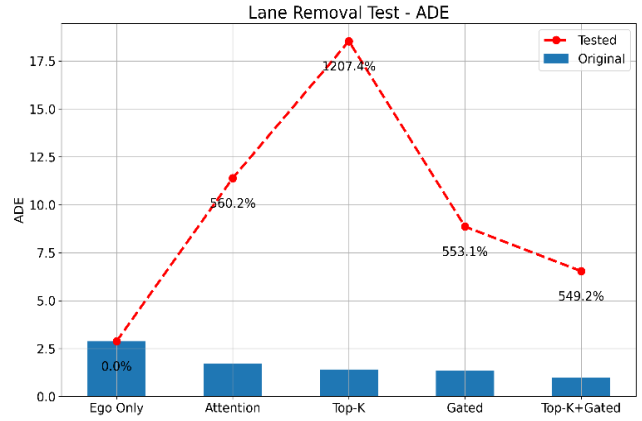


Figure 19: Comparing degradation of ADE during lane context ablation

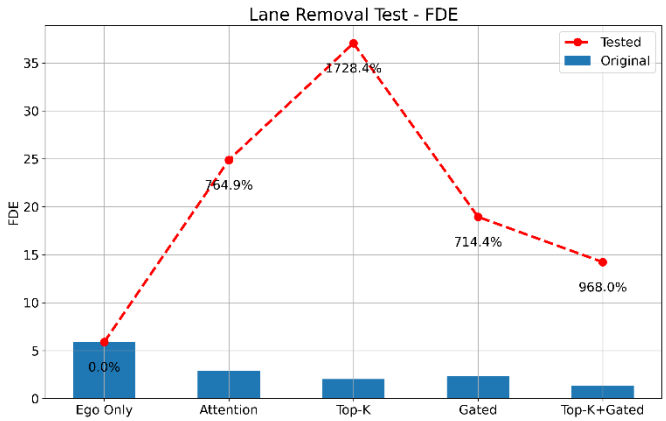


Figure 20: Comparing degradation of FDE during lane context ablation

The degradation of ADE and FDE due to removal of lane context can be analyzed from figures 19 and 20. In the figure, the original values of the evaluation metrics are represented in bar graphs, while the tested values after ablation are represented in dotted line graphs. The degradation values are also given as percentages along with the data point markers

Since the Ego-only trajectory prediction does not use any lane information, the absence of lane context would naturally not cause any changes. All other models show significant degradation, with the order from the least ADE/FDE to the highest being **Top-K+Gated > Gated > Attention > Top-K**. Top-K also shows the highest degradation, making it the most dependent on lane geometry and consequently, more vulnerable to faulty predictions due to degraded lane context. While the degradation for Top-K+Gated is lowest for ADE, the difference is not that far from Attention and Gated Predictors. It also has the second highest FDE metric among the four non-Ego only models.

Such significant degradation along with the poor performances of Top-K and Top-K + Gated predictors, both that use aggressive sparsification, without lane context suggests that just social clues from neighbor interaction representation was not sufficient for proper trajectory reasoning, especially for long-horizon directional forecasting.

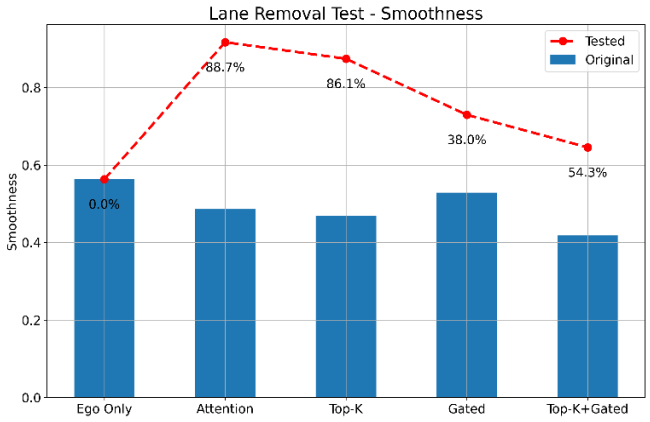


Figure 21: Comparing degradation of Smoothness during lane context ablation

Figure 21 plots the degradation of Smoothness instead, and it shows less degradation compared to ADE/FDE metrics. This indicates that the missing lane context generated inaccurate predictions, but the trajectories by themselves were physically plausible and dynamically coherent. So the models, especially those with lower Smoothness degradations relied on lane context for spatial precision and not motion realism.

The Top-K + Gated predictor retains the lowest Smoothness value from among the non-Ego only models, while the Gated Predictor has the lowest degradation. The

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

other two have higher values and degradation. This indicates that the models that use gating mechanism have more stable trajectory changes even without lane context than models that assign equal importance to all neighboring agents.

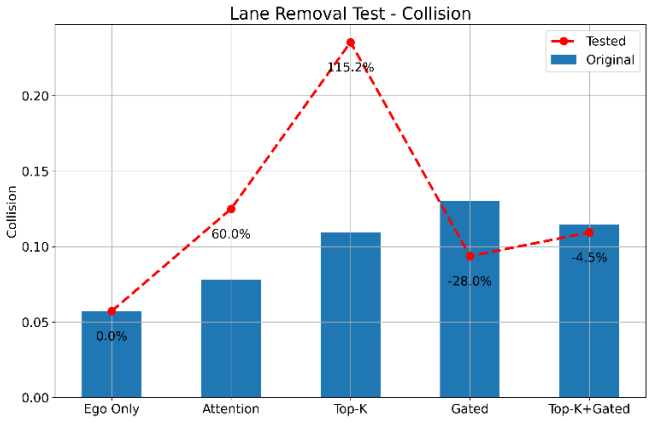


Figure 22: Comparing degradation of Collision Rate during lane context ablation

The figure for collision rate comparison, figure 22, also highlights similar result when considering the order of degradation. Top-K predictor and Attention-based predictors show high collision rate degradation, while the Gated and Top-K+Gated predictors show negative degradation. Such negative degradation shows that lane ablation instead helped reduce unnecessary collisions, possibly by suppressing trajectory exploration to unnecessary close lanes, and providing a more conservative trajectory prediction.

In summary, the Gated Predictor showed the best resilience among all the models when considering degraded lane context, proving the effectiveness of physics-based learning in providing missing lane context. The Top-K and Attention-based predictors both have inherent issues that does not provide such assistance. For instance the equal attention for all the agents in Attention-based and for all the critical agents in Top-K predictors does not help generalize any semantic inferences. However, for the Gated Predictor, the model may be able to generalize that the areas with critical agents is a drivable area while areas with low-importance agents is not, which is more often true in a realistic scenario.

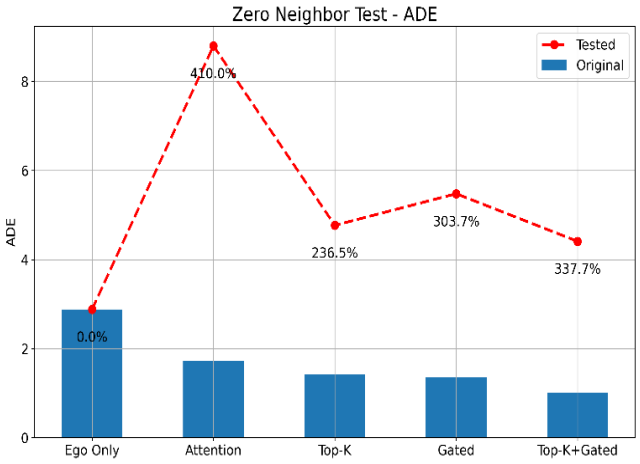
While the proposed hybrid model experienced more degradation, the absolute values for all these error metrics were still the lowest. This proves that while the hard sparsification due to Top-K selection made it difficult for the model to understand spatial boundary reasoning from just the critical neighboring agents, the use of learned gating still compensated enough to generate the best results even with high degradation. So the proposed model achieves the most balanced combination of accuracy, stability and safety under lane context removal.

b) Interaction Ablation: Zero-Neighbor Setting

This study analyzes results from the Zero Neighbor Test that evaluates the degradation of the models by removing all neighbors, and hence all interaction context. In the study, both the absolute tested values of the evaluation metrics (ADE, FDE, Smoothness, Collision Rate), and their degradation from the original baseline is compared. The actual values represents the operational capability of the models under degraded conditions, and the degradation evaluates the dependency of the model on social interaction dynamics represented by the model.

The degradation of ADE and FDE due to removal of neighbors can be analyzed from figure 23. In the figure, the original values of the evaluation metrics are represented in bar graphs, while the tested values after ablation are represented in dotted line graphs. The degradation values are also given as percentages along with the data point markers.

As for individual models, the Ego-only trajectory prediction does not show any changes as expected. However, all other models show significant degradation, with the order from the least ADE/FDE to the highest being Top-K+Gated > Top-K > Gated > Attention. Attention-based Predictor also shows the highest degradation, which makes sense as assigning equal importance to all the neighbors make absence of any neighbor more apparent, let alone all of them. As a consequence, the model is also the most vulnerable to faulty predictions due to degraded neighbor context.



International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

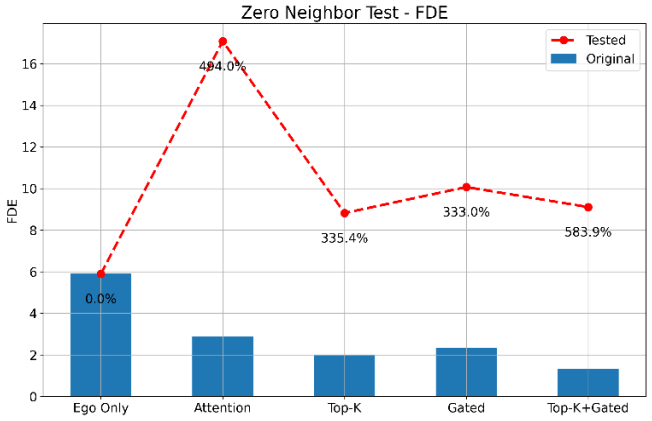


Figure 23: Comparing degradation of ADE and FDE during interaction ablation

Top-K, Gated and Top-K+Gated predictor show similar level of lower ADE/FDE values with comparatively lower degradation. This shows that sparsification or using learned importance reduced the dependency of the model on the interaction context, consequently leading to better generalization of ego trajectory and lane characteristics, which is especially true for structural sparsification, as Top-K and Top-K + Gated models maintain the lowest values.

Among these three Top-K Predictor shows the least degradation and Top-K + Gated shows the highest.

The overall ADE/FDE degradation percentages for this test is significantly less than those for the lane removal test. This indicates that all the models rely more on lane context compared to interaction awareness. However, the values are still somewhat high, which indicates that just lane context is not enough for proper trajectory prediction without social context in a dynamic traffic environment.

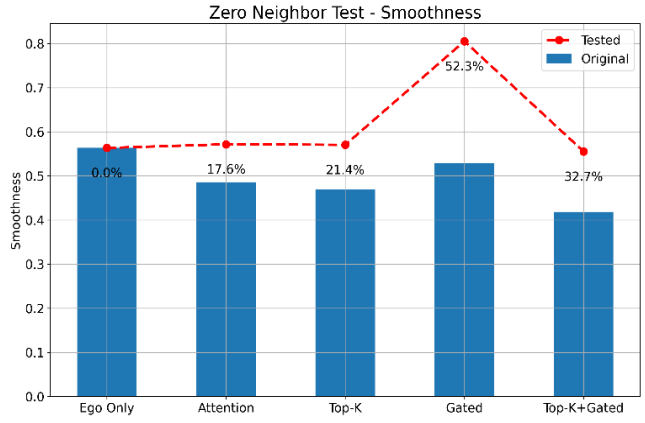


Figure 24: Comparing degradation of Smoothness during interaction ablation

Figure 24 plots the degradation of Smoothness instead, and it shows significantly less degradation compared to ADE/FDE metrics. This indicates that while the degraded interaction context lead to inaccurate trajectory predictions, their motion paths were physically plausible. Combined with Smoothness comparison during the lane removal test, it can be concluded that the ego trajectory motion itself provides enough motion context for realistic and smooth trajectory transitions due to the inclusion of all positional, velocity, acceleration and orientation/directivity information in the ego trajectories.

For individual analysis, the proposed Top-K + Gated predictor retains the lowest Smoothness value from among the non-Ego only models, while the Gated Predictor has the highest. The other two have nearly the same values as that of the proposed model. However, when considering degradation, the other two, Attention-based and Top-K predictors show the lowest percent change, and the Gated predictor shows the highest. This indicates that for models without learned gating, assigning equal importance to all considered nearby agents caused the model to rely less on interaction context for temporal stability. On the other hand, the varying learned scores for different agents based on criticality on gating-based models increased their dependence on the interaction information for precise trajectory changes and temporal motion stability.

Unlike the case for lane context ablation, the collision plot (figure 25) for interaction ablation shows significant degradation for all models, which is the expected behavior as this metric is almost entirely affected by the nearby agents.

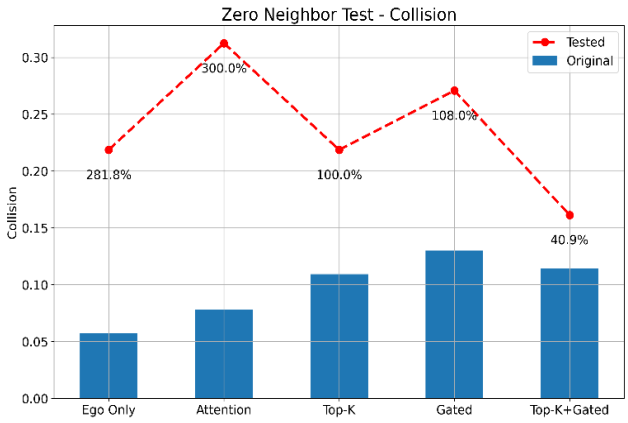


Figure 25: Comparing degradation of Collision Rate during interaction ablation

This is the only case where the Ego-Only agent experiences degradation, indicating that although the trajectories remained unchanged due to absence of any interaction characteristics during training, the model could not generate proper reasoning.

The Attention-based Predictor shows the highest collision and degradation, as it is the model that considers all

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 6, pp 1-25, June-2026

https://doi.org/10.47001/IRJIET/2026.106001

neighboring agents equally. This indicates that this model relied most on motion cues from nearby agents for safe motion behavior. The proposed Top-K + gated model has both the lowest absolute values and the least degradation, making it the safest model even with degradation of social cues.

In summary, the Gated Predictor showed the best resilience among all the models when considering degraded information context, especially when considering long-term accuracy and safety. This demonstrates the efficiency of physics-based learning in compensating for incomplete social interaction information. The Attention-based predictor shows the worst performance without such context for both accuracy and safety. Top-K and Gated Predictors, still maintain respectable performance for all considered metrics, though they are not able to generalize as well as the proposed model (Top-K + Gated). So the proposed model achieves the most balanced combination of accuracy, stability and safety under zero neighbor robustness test.

c) Stochastic Perturbation Sensitivity Analysis

Stochastic Perturbation Sensitivity Analysis evaluates the performance of models on incremental noise conditions. In this study, Gaussian noise was introduced with increasing standard deviation of 0.5, 1.0 and 2.0, and the resulting variations in the evaluation metrics can be analyzed from figures 26, 27 and 28.

Figure 26 illustrates the effect of noise added to input ego trajectory data on the ADE and FDE evaluation criteria.

According to the figure, the Ego-Only model shows the highest sensitivity to positional noise at all levels. This shows that just using trajectory information as input leaves no room for any error, as the model lacks any contextual mechanism to compensate for degraded temporal and positional observations.

The Top-K and Attention-based predictors actually low degradation at all noise levels, especially at higher perturbation, proving that equally distributed contextual aggregation helps suppress errors due to positional sensing noise. While their values at low standard deviation Gaussian noise level are higher than the hybrid model, they have the flattest slope, indicating the most resilience to noise. The Gated and Top-K + Gated predictors show more degradation in comparison to their baseline at all noise levels in comparison. This indicates that while learned gating provides the best motion reasoning, it also increases the model's dependency on accuracy of temporal observations.

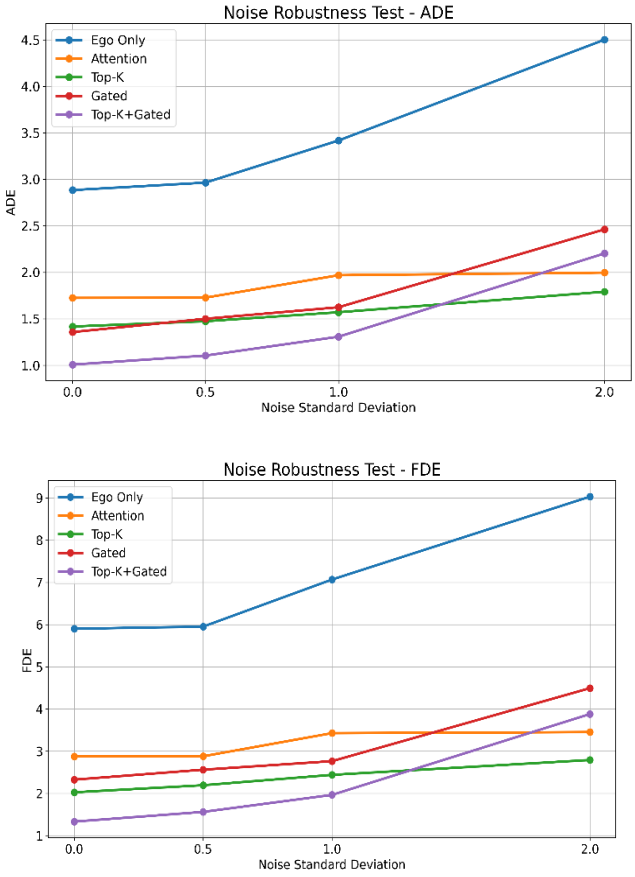


Figure 26: Stochastic Perturbation Sensitivity Analysis based on ADE and FDE

When comparing change in Smoothness through figure 27, it can be seen that the trajectory motion continuity doesn't get affected as much due to incremental noise. All the models retain not only their low range of Smoothness, but also their respective order as shown in the figure.

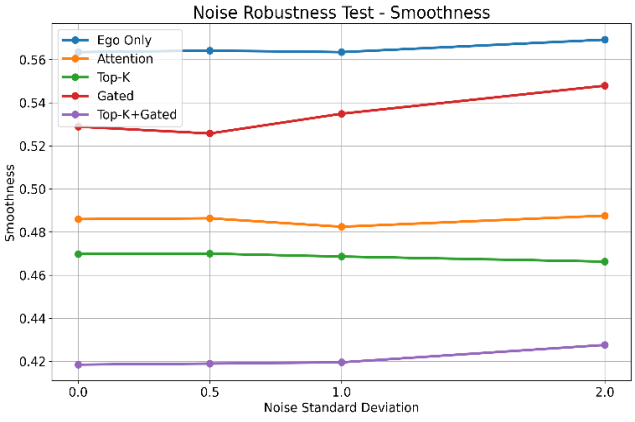


Figure 27: Stochastic Perturbation Sensitivity Analysis based on Smoothness

The case for collision is also more or less similar to that for Smoothness, as shown in figure 28.

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

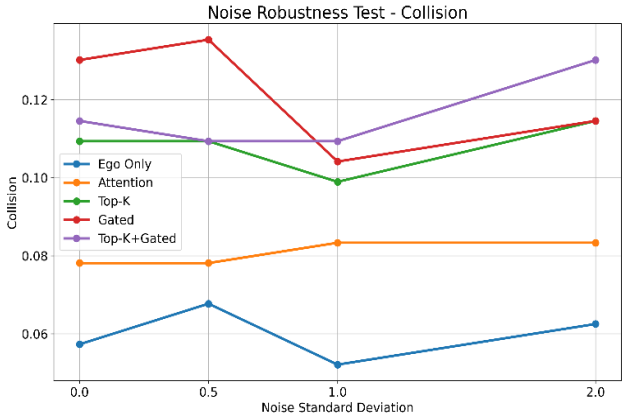


Figure 28: Stochastic Perturbation Sensitivity Analysis based on Collision Rate

While the respective order for models with the highest to the lowest collision rate experiences some changes, the range of collision rate is maintained at 0 – 0.2 throughout the noise perturbation levels, indicating that noisy positional sensors has limited effect on the collision rate of the predicted trajectories from all models.

III. LIMITATIONS AND FUTURE AVENUES

The results also highlight certain limitations within the model architecture. The major ones include:

- Due to lack of multimodal trajectory prediction, only the trajectory of the ego vehicle were predicted. Implementation of multi-agent trajectory prediction of critical agents along with the ego vehicle within reasonable computational overhead can be considered for future studies.
- The system lacks any consideration for uncertain neighboring agent predictions. Even the ego trajectory uses deterministic modeling, making it difficult to adapt to uncertain and dynamic situations like pedestrians walking on the sidewalk suddenly jumping into the path of the ego vehicle. Implementation of mimimum-K ADE (minADE) and minimum-K FDE (minFDE) using a gaussian distribution during evaluation may help create a basic uncertainty trajectory model, but better architecture needs to be studied even for lightweight models.
- Lane importance currently depends solely on lane geometry. While it gives reliable prediction on straight and curved lanes, it is not very suitable for predicting trajectories at intersections. A better model would be to condition the importance on ego trajectory, such as: $s_j = MLP([h_{ego}, h_{lane,j}])$ instead of $s_j = W.h_{lane,j}$
- Interaction and map usage has been kept simple for computational convenience. However, lack of complex interaction mapping and map encoding also means there is an upper limit to how precise the model can determine future trajectories. This limitation is inherent to any such lightweight models, and may only be slowly addressed in the future.

Further improvements to the model comes with having to add more computational power and may not support real-time prediction. However, further studies on multi-modal prediction, implementation of graph-based interaction, and better map representation are all possible avenues even for efficient and timely trajectory prediction.

V. CONCLUSION

In conclusion, this study proves the feasibility of using physics-guided sparsification for a more efficient trajectory prediction. Integration of both physics metrics to reduce computational overhead and non-linear scored interaction learning using multi-layer perceptron to represent latent relation between interacting agents has proved effective on multiple levels instead of applying same attention to all the agents.

REFERENCES

[1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems." *Transactions of the ASME–Journal of Basic Engineering*, 82(1), 35-45, 1960.

[2] Zhang, J., & Cho, Y., "Deep Trajectory Prediction: A Deep Learning Approach." *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[3] Gupta, A., & Khan, M., "Predicting Future Trajectories with Generative Adversarial Networks." *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2018.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[5] Z. Sheng, Z. Huang, and S. Chen, "EPG-MGCN: Ego-Planning Guided Multi-Graph Convolutional Network for Heterogeneous Agent Trajectory Prediction," *arXiv preprint arXiv:2303.17027*, 2023.

[6] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, "MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 6, pp. 4115–4128, 2024.

[7] Z. Zhao, C. Hua, F. Berto, K. Lee, Z. Ma, J. Li, and J. Park, "Trajevo: Designing Trajectory Prediction Heuristics via LLM-Driven Evolution," *arXiv preprint* arXiv:2505.04480, 2025.

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

[8] C. Liu, S. He, H. Liu, and J. Chen, "Intention-Aware Denoising Diffusion Model for Trajectory Prediction," *arXiv preprint* arXiv:2403.09190, 2024.

[9] Y. Tang, H. He, Y. Wang, and Y. Wu, "Using a Diffusion Model for Pedestrian Trajectory Prediction in Semi-Open Autonomous Driving Environments," *IEEE Sensors Journal*, vol. 24, no. 10, pp. 17208–17218, 2024.

[10] H. Sun, Z. Zhao, and Z. He, "Reciprocal Learning Networks for Human Trajectory Prediction," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7416–7425, 2020.

[11] K. Saleh, "Pedestrian Trajectory Prediction for Real-Time Autonomous Systems via Context-Augmented Transformer Networks," *Sensors*, vol. 22, no. 19, Art. no. 7495, Oct. 2022. doi:10.3390/s22197495

[12] X., et al., "Context CVGN: A conditional multimodal trajectory prediction network based on scene semantic modeling," *Information Sciences*, vol. 666, May 2024, Art. no. 120433.

[13] B. Dong, H. Liu, Y. Bai, J. Lin, Z. Xu, X. Xu, and Q. Kong, "Multi-modal Trajectory Prediction for Autonomous Driving with Semantic Map and Dynamic Graph Attention Network," *NeurIPS Workshop: Machine Learning for Autonomous Driving*, Vancouver, Canada, 2020. arXiv:2103.16273

[14] C. Ji, Q. Li, "TrajSceneLLM: A Multimodal Perspective on Semantic GPS Trajectory Analysis," *arXiv preprint*, Jun. 2025. arXiv:2506.16401

[15] Z. Sun, Z. Wang, L. Halilaj, and J. Luettin, "SemanticFormer: Holistic and semantic traffic scene representation for trajectory prediction using knowledge graphs," *IEEE Robotics and Automation Letters*, vol. 9, pp. 7381–7388, 2024.

[16] H. Zhang et al., "Probabilistic semantic mapping for autonomous driving in urban environments," *Sensors*, vol. 23, no. 14, Art. no. 6504, 2023. doi: 10.3390/s23146504.

[17] Z. Li and H. Yu, "Trajectory prediction for autonomous driving using a transformer network," *arXiv preprint*, 2024. arXiv:2402.16501

[18] X. Wang et al., "IAtraj: Multi-modal trajectory prediction through contextual information spatio-temporal interaction and awareness," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 9, no. 5, pp. 17–28, 2024. doi: 10.9781/ijimai.2024.09.001.

[19] J. Zhang, Y. Fan, F. Hui, E. Tan, and X. Zhou, "Interaction-aware trajectory prediction for heterogeneous agents in shared spaces," *Physica A: Statistical Mechanics and its Applications*, 2025. Art. no. 131054.

[20] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-WaGDAT: Interaction-aware trajectory prediction via Wasserstein graph double-attention network," *arXiv preprint*, 2020.arXiv:2002.06241

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[22] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, doi: 10.1207/s15516709cog1402_1.

[23] K. Cho *et al*., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.

[24] A. Vaswani *et al*., "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 6000–6010, doi: 10.48550/arXiv.1706.03762.

[25] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020, doi: 10.48550/arXiv.2006.11239.

[26] X. Mo, Y. Xing, and C. Lv, "Interaction-Aware Trajectory Prediction of Connected Vehicles using CNN-LSTM Networks," 2020.

[27] Y. Peng, G. Zhang, J. Shi, B. Xu, and L. Zheng, "SRAI-LSTM: A Social Relation Attention-based Interaction-aware LSTM for Human Trajectory Prediction," *Neurocomputing*, vol. 490, pp. 258–268, 2022.

AUTHORS BIOGRAPHY

Abhishek Silwal, Pursuing Msc in Computer Engineering Specialization in Data Science and Analytics at Pulchowk Campus, IOE, Kathmandu, Nepal.

ISSN (online): 2581-3048
Volume 10, Issue 6, pp 1-25, June-2026
https://doi.org/10.47001/IRJIET/2026.106001

Asst. Prof. Anku Jaiswal, at Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Kathmandu, Nepal.

Citation of this Article:

Abhishek Silwal, & Anku Jaiswal. (2026). Physics-Guided Interaction Sparsification for Efficient Ego Trajectory Prediction. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(6), 1-25. Article DOI https://doi.org/10.47001/IRJIET/2026.106001

****

