PROBING INTERACTION AWARENESS IN TRAJECTORY PREDICTION WITH COUNTERFACTUAL PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory prediction models for autonomous driving often incorporate neighboring-agent information through social pooling, graph-based interaction modeling, message passing, and attention mechanisms. However, commonly used evaluation metrics such as Average Displacement Error (ADE) and Final Displacement Error (FDE) do not reveal whether predictions are actually influenced by surrounding agents. To address this limitation, we introduce a counterfactual sensitivity framework that directly tests interaction dependence by perturbing neighboring-agent trajectories while keeping the target agent unchanged. We apply three perturbation strategies agent removal, spatial displacement, and Gaussian noise to evaluate how predictions respond to controlled changes in neighboring agents. We evaluate LSTM-Interaction, AgentFormer, Trajectron++, and FIERY on the nuScenes dataset using four sensitivity metrics, including the proposed Zero-Reaction Rate (ZRR). The results show that most models exhibit limited response to neighboring-agent perturbations, particularly under realistic geometric and noise-based modifications. Across the evaluated models, Zero-Reaction Rates range from 65% to 99%, indicating that many perturbations produce little or no change in the predicted trajectory. We also observe that sensitivity decreases as scene density increases, suggesting that interaction information may be used less effectively in crowded environments. These findings demonstrate why counterfactual perturbation is necessary as a complementary evaluation tool: prediction accuracy alone does not establish whether a model meaningfully uses neighboring-agent information. Counterfactual sensitivity analysis therefore provides additional insight into how trajectory prediction models utilize interactions.

Keywords: Trajectory Prediction, Autonomous Driving, Multi-Agent Interaction, Counterfactual Analysis, Interaction Awareness, Sensitivity Analysis

1 INTRODUCTION

Trajectory prediction is a key component of autonomous driving systems, enabling the ego vehicle to anticipate the future motions of surrounding traffic participants and plan safe maneuvers accordingly. Given the observed motion histories of agents in a scene, a prediction model must estimate their future trajectories while accounting for interactions with nearby vehicles, pedestrians, and other dynamic actors. To address this challenge, recent methods have incorporated increasingly sophisticated interaction modeling mechanisms. Social-LSTM (Alahi et al., 2016) introduced occupancy-based social pooling, Social-GAN (Gupta et al., 2018) extended this idea through generative modeling, graph-based approaches such as STGAT (Huang et al., 2019) explicitly model relationships among agents, and transformer-based architectures such as AgentFormer (Shi et al., 2021) leverage attention mechanisms to capture complex scene interactions.

Although these methods have achieved significant improvements on standard benchmarks, their evaluation remains largely centered on prediction accuracy metrics such as Average Displacement Error (ADE) and Final Displacement Error (FDE). While these metrics measure the discrepancy between predicted and ground-truth trajectories, they do not reveal whether the model actually utilizes information from surrounding agents when generating predictions. As a result, a model may achieve

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

competitive ADE and FDE scores while relying primarily on the motion history of the target agent. Such behavior can be problematic in complex autonomous driving scenarios where future motion is often influenced by interactions with neighboring traffic participants.

During preliminary experiments, we observed that modifying neighboring-agent trajectories did not always lead to noticeable changes in model predictions, even when the perturbations were substantial. This observation motivated the question we investigate here: *to what extent do modern trajectory prediction models rely on neighboring-agent information when generating future trajectory predictions?*

To investigate this question, we introduce a **counterfactual sensitivity framework** for evaluating interaction awareness in trajectory prediction models. Rather than focusing solely on prediction accuracy, the proposed framework systematically perturbs neighboring-agent trajectories while keeping the target agent unchanged and analyzes the resulting changes in predicted trajectories. The intuition is straightforward: *If a model effectively incorporates interaction information, changes in neighboring agents should influence its predictions.* Conversely, limited response to such perturbations may indicate weak dependence on interaction cues.

We apply the proposed framework to four representative trajectory prediction architectures spanning recurrent, graph-based, transformer-based, and occupancy-based paradigms. All models are evaluated on the nuScenes dataset (Caesar et al., 2020) under a unified protocol. We introduce controlled counterfactual perturbations agent removal, spatial displacement, and Gaussian noise while keeping the target agent fixed. Four sensitivity metrics, including the proposed **Zero-Reaction Rate (ZRR)**, quantify the magnitude and frequency of prediction changes. This model-agnostic framework enables systematic evaluation of whether different architectures actually utilize neighboring-agent information beyond conventional accuracy metrics.

The main contributions of this work are summarized as follows:

- A unified **counterfactual sensitivity evaluation framework** that assesses interaction awareness through structured perturbations (removal, shift, noise) of neighboring-agent trajectories.
- Four complementary sensitivity metrics, including a novel **Zero-Reaction Rate (ZRR)** that measures the prevalence of prediction insensitivity across a dataset.
- A systematic cross-architecture evaluation of LSTM-Interaction, AgentFormer, Trajectron++, and FIERY under a common protocol on nuScenes.
- Empirical evidence that models with comparable ADE/FDE can exhibit substantially different sensitivity to neighboring-agent perturbations, exposing a limitation of accuracy-only evaluation.

2 Related Work

This section reviews representative trajectory prediction methods and existing counterfactual approaches, highlighting the gap in systematic interaction-sensitivity evaluation.

2.1 Trajectory Prediction Models

Trajectory prediction has evolved from social pooling to graph-based, transformer-based, and occupancy-based interaction modeling. Social-LSTM (Alahi et al., 2016) and Social-GAN (Gupta et al., 2018) model neighbor interactions through social pooling and generative learning, while STGAT (Huang et al., 2019) and GRIP (Li et al., 2019) explicitly encode agent relationships using graph-based representations. AgentFormer (Shi et al., 2021) models cross-agent dependencies through attention, whereas FIERY (Hu et al., 2021) captures interactions implicitly through shared BEV representations.

Despite these advances, existing methods are primarily evaluated using ADE and FDE, which measure prediction accuracy but do not directly establish whether neighboring-agent information influences the prediction. Moreover, their interaction mechanisms differ substantially across architectures, making direct evaluation of interaction sensitivity difficult. This motivates our counterfactual

Table 1: Comparison of existing trajectory forecasting methods according to interaction modeling, counterfactual analysis, and sensitivity evaluation.

Method	IM	CA	SE
Social LSTM (Alahi et al., 2016)	✓	✗	✗
Social GAN (Gupta et al., 2018)	✓	✗	✗
STGAT (Huang et al., 2019)	✓	✗	✗
AgentFormer (Shi et al., 2021)	✓	✗	✗
Trajectron++ (Salzmann et al., 2020)	✓	✗	✗
CausalAgents (Roelofs et al., 2022)	✓	✓	✗
CaDeT (Pourkeshavarz et al., 2024)	✓	✓	✗
Counterfactual Methods	✓	✓	✗
Proposed	✓	✓	✓

IM: Interaction Modeling, CA: Counterfactual Analysis, SE: Sensitivity Evaluation.

framework, which provides a unified protocol for measuring how predictions respond to controlled perturbations of neighboring agents.

Recent trajectory forecasting architectures have further advanced prediction performance through large-scale attention mechanisms and unified modeling frameworks. Examples include Wayformer (Nayakanti et al., 2023), QCNet (Zhou et al., 2023), UniTraj (Shi et al., 2024), and MotionLM (Bhattacharyya et al., 2024). These approaches primarily focus on improving forecasting accuracy and scalability across autonomous-driving benchmarks. Table 1 summarizes existing approaches in terms of interaction modeling, counterfactual analysis, and sensitivity evaluation.

Despite substantial architectural advances, these models are primarily evaluated using accuracy-oriented metrics such as ADE and FDE. While such metrics quantify prediction accuracy, they provide limited insight into how strongly neighboring-agent information influences model predictions. The interaction sensitivity of recent architectures under counterfactual perturbations also remains largely unexplored. In contrast, our work focuses on evaluating interaction sensitivity rather than improving predictive accuracy.

3 METHODOLOGY

This section presents the proposed counterfactual sensitivity evaluation framework. The framework systematically perturbs neighboring-agent trajectories while keeping the target agent unchanged and measures the resulting changes in model predictions. Three perturbation strategies are considered, ranging from structural modifications (agent removal) to realistic geometric and perceptual disturbances (spatial shifts and Gaussian noise). The resulting prediction differences are quantified using four complementary sensitivity metrics and evaluated across multiple trajectory prediction architectures on the nuScenes dataset.

3.1 PROBLEM FORMULATION

Let $\mathbf{X}_T = \{x_T^1, x_T^2, \ldots, x_T^t\}$ denote the observed past trajectory of a target agent over t timesteps, and let $\mathbf{X}_N = \{X_{N_1}, X_{N_2}, \ldots, X_{N_k}\}$ denote the set of past trajectories of k neighboring agents. A trajectory prediction model estimates the future positions of the target agent as:

$$\hat{Y} = f(\mathbf{X}_T, \mathbf{X}_N), \tag{1}$$

where $\hat{Y} = \{y^{t+1}, \ldots, y^{t+H}\}$ represents predicted future positions over a prediction horizon H (we use H, not T, to avoid overloading the target-agent subscript $\mathbf{X}_T$).

While models are designed to leverage $\mathbf{X}_N$ for interaction reasoning, it remains empirically unclear how much $\mathbf{X}_N$ actually influences $\hat{Y}$. We investigate this by measuring how predictions change when $\mathbf{X}_N$ is systematically perturbed while $\mathbf{X}_T$ is held fixed.

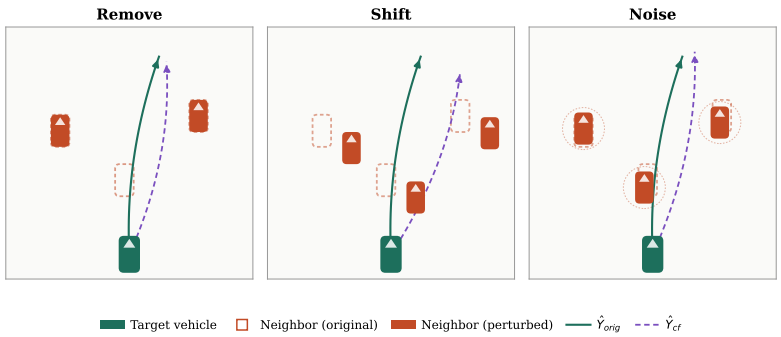

Figure 1: Counterfactual perturbations of neighboring agents: removal, spatial shift, and Gaussian noise, with the target agent fixed.

3.2 Counterfactual Perturbations

Given a perturbed neighbor set $\mathbf{X}'_N = P_p(\mathbf{X}_N)$ under perturbation strategy p, the corresponding counterfactual prediction is:

$$\hat{Y}_{cf} = f(\mathbf{X}_T, \mathbf{X}'_N). \tag{2}$$

For brevity, we write $\hat{Y}_{orig} \equiv \hat{Y}$ (Eq. 1) and $\hat{Y}_{cf}$ (Eq. 2) for the original and counterfactual predicted trajectories of a given sample, each consisting of the H predicted positions $\hat{Y}^{(t+1)}, \ldots, \hat{Y}^{(t+H)}$. As in Algorithm 1, all four sensitivity metrics below are computed per sample and then aggregated over the N evaluation samples.

We define three complementary perturbation types, each targeting a different aspect of neighbor information.

Structural Perturbation (Remove): One or more neighboring agents are completely removed from the scene, i.e., $|\mathbf{X}'_N| < |\mathbf{X}_N|$. This is the strongest structural disruption to the interaction context and is used to assess the maximum response under agent removal. We sweep $r \in \{1, 2, 3\}$ removals to study scaling behavior.

Geometric Perturbation (Shift): All neighbor trajectories are translated by a fixed spatial offset δ meters while preserving their motion dynamics:

$$X'_{N_i} = X_{N_i} + \delta, \quad \forall i \in \{1, \ldots, k\}. \tag{3}$$

We sweep $\delta \in \{1.0, 2.0, 3.0, 4.0\}$ meters. This tests whether models respond to realistic changes in relative agent positioning, such as those caused by upstream localization errors.

Perceptual Perturbation (Noise): Zero-mean Gaussian noise is added independently to each neighbor position at every timestep:

$$X'_{N_i}(t) = X_{N_i}(t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}). \tag{4}$$

We use $\sigma \in \{0.5, 1.0\}$ meters, reflecting typical localization uncertainty in real-world autonomous driving systems.

These perturbation types span a spectrum from extreme structural changes (agent removal) to subtle realistic degradation (noise), allowing characterization of the full sensitivity profile of each model.

3.3 Sensitivity Metrics

We define four metrics to quantify different aspects of model response to counterfactual perturbations. Subscript n indexes the evaluation sample and superscript $(t+h)$ indexes the timestep within a predicted trajectory.

(1) Trajectory Sensitivity (S): The average pointwise deviation between original and counterfactual predicted trajectories, per sample n:

$$S_n = \frac{1}{H} \sum_{h=1}^{H} \left\| \hat{Y}_{orig,n}^{(t+h)} - \hat{Y}_{cf,n}^{(t+h)} \right\| \tag{5}$$

We report $S = \frac{1}{N} \sum_{n=1}^{N} S_n$, the mean over all samples. High S indicates stronger response to perturbations.

(2) Final Displacement Shift (S_{final}): The deviation at the final prediction timestep, per sample n:

$$S_{final,n} = \left\| \hat{Y}_{orig,n}^{(t+H)} - \hat{Y}_{cf,n}^{(t+H)} \right\| \tag{6}$$

reported as $S_{final} = \frac{1}{N} \sum_{n=1}^{N} S_{final,n}$. This captures whether long-horizon predictions are affected, which is particularly relevant for planning safety.

(3) Zero-Reaction Rate (ZRR): The proportion of samples for which the model exhibits negligible response, using the full H-step trajectory per sample:

$$\text{ZRR} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1} \left[\left\| \hat{Y}_{orig,n} - \hat{Y}_{cf,n} \right\| < \tau \right] \tag{7}$$

where $\hat{Y}_{orig,n}$ and $\hat{Y}_{cf,n}$ denote the concatenated H-step trajectory for sample n, $\tau = 0.05$ m is the sensitivity threshold, and N denotes the number of evaluation samples. We set τ to a small value to exclude insignificant prediction variations.

ZRR directly measures the prevalence of insensitivity across the dataset, complementing the average-based metrics above.

(4) Error Amplification (ΔE): The change in prediction error with respect to ground truth after perturbation:

$$\Delta E = E_{cf} - E_{orig} \tag{8}$$

where E denotes ADE against ground truth. If $\Delta E \approx 0$ despite a large perturbation, the model's accuracy is largely unaffected by neighbor changes, suggesting limited reliance on neighbor context for ground-truth-aligned prediction. Algorithm 1 in Appendix A summarizes the full evaluation procedure.

4 Models Under Evaluation

We evaluate four architectures spanning distinct interaction-modeling paradigms, from a simple recurrent baseline to an occupancy-based map predictor, Table 2 summarizes their paradigm, interaction mechanism, and output space.

LSTM-Interaction (Alahi et al., 2016) Our recurrent baseline encodes the target agent's and each neighbor's past trajectory with shared-weight LSTM encoders, fuses the resulting hidden states $h_T, h_N \in \mathbb{R}^d$ by concatenation, and decodes future positions with an MLP. Its interaction mechanism is explicit and simple, making sensitivity results straightforward to interpret.

AgentFormer (Shi et al., 2021) applies agent-aware transformer attention jointly over all agents and timesteps, encoding agent identity into positional embeddings so each token can attend to *which* agent a feature belongs to across the whole scene. We use the publicly released nuScenes pre-trained checkpoint without modification.

Trajectron++ (Salzmann et al., 2020) is a probabilistic graph-based predictor that builds a dynamic interaction graph at each observation timestep, with learned edge weights representing pairwise

Table 2: Representative Characteristics of Evaluated Models

Model	Paradigm	Interaction Modeling	Output Space
LSTM-Interaction	Recurrent	Social Pooling	Trajectory
AgentFormer	Transformer	Cross-Agent Attention	Trajectory
Trajectron++	Graph-Based	Learned Interaction Graph	Trajectory
FIERY	Occupancy-Based	Shared Spatial Features	Occupancy Map

influence, and propagates relational features via graph convolution before decoding a distribution over future trajectories.

FIERY (Hu et al., 2021) takes a fundamentally different approach: rather than predicting per-agent positions explicitly, it lifts surround-view camera features into a bird's-eye-view (BEV) representation and predicts future instance segmentation maps for all agents jointly. Interactions are encoded *implicitly* through shared spatial features, making FIERY a strong test case since its interaction mechanism is not separately inspectable. We evaluate using the released nuScenes occupancy weights and derive ADE/FDE from predicted instance centroids.

5 Experimental Setup

This section describes the dataset, evaluated models, implementation settings, and evaluation protocol used for counterfactual sensitivity analysis. All models are evaluated under a common inference-time perturbation protocol to ensure consistent comparison across architectures.

5.1 Dataset

All experiments are conducted on the **nuScenes** dataset (Caesar et al., 2020), a large-scale autonomous driving benchmark comprising 1,000 driving scenes collected in Boston and Singapore. Each scene provides 20 seconds of data at 2 Hz with full sensor coverage (6 cameras, 1 LiDAR, 5 RADAR) and 3D bounding box annotations for all agents. We use the standard train/val splits with 700 and 150 scenes respectively. All trajectories are normalized relative to each target agent's initial position and heading. To analyze the effect of crowd complexity, scenes are stratified into three density levels based on the number of neighboring agents visible to the target: **Low** ($\leq$ 2 agents), **Medium** (3–5 agents), and **High** ($>$ 5 agents). These thresholds follow density-based stratification conventions used in prior trajectory prediction studies and yield a reasonably balanced sample distribution across groups in the nuScenes validation set.

5.2 Experimental Environment

Experiments are conducted on an NVIDIA RTX A5000 GPU using PyTorch. The evaluation environment uses CUDA-enabled inference with all models evaluated under the same computational setup. Perturbations are applied only during inference without retraining or fine-tuning.

5.3 Models

The four evaluated models represent distinct trajectory prediction paradigms: recurrent interaction modeling (LSTM-Interaction), transformer-based interaction modeling (AgentFormer), graph-based interaction modeling (Trajectron++), and occupancy-based prediction (FIERY). Detailed architectural descriptions are provided in Section 4.

5.4 Implementation Details

The LSTM-Interaction model is implemented in PyTorch, trained with the Adam optimizer (lr = 10^{-3}, batch size = 32) for 50 epochs. AgentFormer and Trajectron++ are evaluated using their official pre-trained checkpoints without any fine-tuning. FIERY is evaluated using its released nuScenes weights. All counterfactual perturbations are applied at inference time only, no model is retrained or fine-tuned during evaluation.

Table 3: Baseline prediction accuracy on the nuScenes validation set.

Model	ADE (m)	FDE (m)
LSTM-Interaction	0.891	2.007
AgentFormer	1.856	3.889
Trajectron++(1s)	0.188	2.782
Trajectron++(4s)	3.301	5.328
FIERY	1.618	1.977

Table 4: Sensitivity scaling under progressive neighbor removal (LSTM-Interaction).

Removed (k)	S	S_{final}	ZRR (%)
1	0.0572	0.1046	87.85
2	0.1001	0.1838	78.74
3	0.1317	0.2414	70.54

5.5 Evaluation Metrics

For trajectory-space models (LSTM-Interaction, AgentFormer, Trajectron++) we report: ADE (Average Displacement Error), FDE (Final Displacement Error), and the four sensitivity metrics defined in Section 3. For FIERY, which operates in occupancy space, we additionally report Intersection-over-Union (IoU), Panoptic Quality (PQ), Segmentation Quality (SQ), and Recognition Quality (RQ).

6 Results

This section presents the baseline prediction performance and counterfactual sensitivity results. We analyze how different perturbations affect model predictions and examine sensitivity across perturbation magnitude and scene density.

6.1 Baseline Prediction Accuracy

Before evaluating counterfactual sensitivity, we first establish the baseline prediction performance of each model on the nuScenes validation set. Table 3 reports the prediction accuracy obtained using the original, unperturbed trajectories. These results correspond to the zero-perturbation setting (i.e., no agent removal, no spatial displacement, and no injected noise) and therefore serve as the reference point for all subsequent sensitivity analyses. The results confirm that the evaluated implementations produce reasonable prediction performance under standard evaluation conditions and provide a suitable foundation for sensitivity analysis. Since the objective of this work is not to compare predictive accuracy, but rather to investigate how neighboring-agent perturbations influence model behavior, the values in Table 3 are primarily used as reference measurements against which all counterfactual experiments are evaluated.

6.2 Scaling Behavior Under Neighbor Removal

To study the relationship between perturbation magnitude and model response, we progressively remove $k \in \{1, 2, 3\}$ neighboring agents (Table 4). Both S and S_{final} increase monotonically with k, and ZRR decreases correspondingly. However, even after removing three neighbors, ZRR remains at 70.54%, indicating that the majority of predictions still do not change. This confirms that the model requires large structural disruptions to exhibit any meaningful response.

6.3 Counterfactual Sensitivity: AgentFormer

Table 5 shows that AgentFormer is considerably more sensitive to spatial perturbations than LSTM-Interaction. ADE increases by $2.1\times$ at a 0.5 m perturbation and by $7.9\times$ at 5 m, suggesting that the attention mechanism encodes relative spatial positions strongly. However, this rapid degradation indicates sensitivity to *positional encoding* rather than semantic interaction reasoning: the model's predictions collapse as soon as neighbor coordinates deviate from training distribution, rather than gracefully adapting to changed social context.

Table 5: AgentFormer ADE/FDE under spatial perturbation of neighbor positions.

Perturbation (m)	ADE (m)↓	FDE (m)↓
0	1.856	3.889
0.5	3.960	6.134
1.0	5.200	7.637
2.0	7.350	9.760
5.0	14.671	16.860

Table 6: Trajectron++ ADE under fixed and random counterfactual perturbations.

Horizon	Original	Fixed CF	Random CF	$\lvert\Delta\rvert$
1s	0.188	0.188	0.188	0.0004
2s	0.742	0.742	0.742	0.0004
3s	1.777	1.777	1.775	0.0023
4s	3.301	3.301	3.303	0.0020

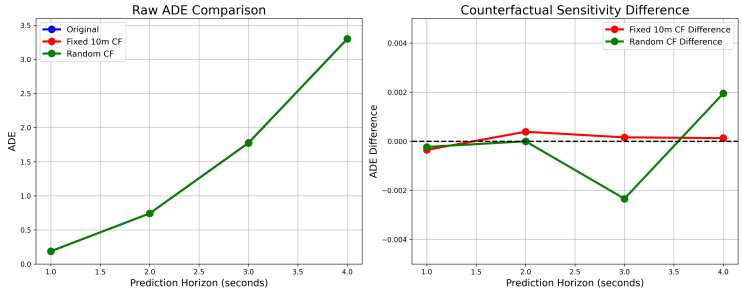

Figure 2: Effect of increasing neighbor perturbations on FIERY. ADE increases substantially while occupancy metrics degrade with perturbation magnitude.

6.4 Counterfactual Sensitivity: Trajectron++

Trajectron++ exhibits the lowest sensitivity of all evaluated models (Table 6). ADE differences under both fixed 10 m and random counterfactual perturbations remain below 0.003 m across all prediction horizons from 1s to 4s. Even at the 4 second horizon where absolute ADE reaches 3.301 m, the counterfactual difference is at most 0.002 m - less than 0.07% of the original error.

6.5 Counterfactual Sensitivity: FIERY

FIERY is markedly more sensitive to neighboring-agent perturbations than the trajectory-space models (Table 7): shifting neighbors from 0 m to 10 m raises ADE from 1.618 m to 3.692 m and FDE from 1.977 m to 4.261 m, while Recognition Quality collapses from 59.3 to 14.1 and Panoptic Quality from 44.0 to 10.0, yet Segmentation Quality stays nearly flat (74.3 $\rightarrow$ 70.8), since PQ = SQ$\times$RQ, this decomposition localizes the degradation almost entirely to instance-association failures – agents being lost, merged, or mismatched across frames, rather than to degraded mask quality for instances the model still correctly identifies, indicating that FIERY's implicit BEV interaction encoding is vulnerable specifically at the agent-recognition stage, not at shape or trajectory regression.

6.6 Density, Failure-Mode, and Error-Amplification Analysis (LSTM-Interaction)

Scene density. Table 8 reveals a counter-intuitive trend: sensitivity decreases substantially as scene density increases, from 0.2218 in low-density scenes to just 0.0002 in high-density scenes, over three orders of magnitude. Rather than benefiting from additional interaction information, the model appears to rely more heavily on the target agent's own motion history as scenes get more crowded.

Failure distribution. Table 9 reports prediction responses across all perturbations: no detectable change occurs in 65.08% of samples, a further 25.75% show only minor deviations, and just 9.17%

Table 7: FIERY performance under increasing neighbor perturbations.

Shift	ADE	FDE	IoU	PQ	SQ	RQ
0 m	1.618	1.977	57.8	44.0	74.3	59.3
1 m	1.800	2.139	39.9	13.0	65.5	19.8
5 m	3.006	3.368	11.8	9.1	70.3	13.0
10 m	3.692	4.261	11.0	10.0	70.8	14.1

Table 8: Sensitivity by scene density.

Density	Sens.
Low ($\leq$2)	0.2218
Med. (3–5)	0.0007
High ($>$5)	0.0002

show medium or large changes confirming that limited response is common across the evaluation set rather than confined to a few scenarios.

Error amplification. Table 10 reports ΔE under increasing neighbor-shift perturbations, unlike S, which measures prediction change directly, ΔE tests whether that change affects accuracy relative to ground truth. It stays extremely close to zero (and slightly negative) even at a 15 m shift, where the change in ADE is under 4×10^{-5} m ,negligible relative to typical errors. Together with the sensitivity and ZRR results, this indicates the model produces nearly identical outputs and essentially unchanged accuracy despite substantial modifications to surrounding-agent trajectories.

7 Discussion

This study introduced a counterfactual perturbation framework for evaluating the interaction sensitivity of trajectory prediction models. The results reveal substantial variation in how neighboring-agent information influences predictions. LSTM-Interaction and Trajectron++exhibit minimal sensitivity to perturbations, whereas AgentFormer and FIERY demonstrate stronger dependence on interaction information. These findings indicate that conventional accuracy metrics alone are insufficient for understanding how models utilize neighboring-agent context.

7.1 Cross-Architecture Comparison

The evaluated architectures exhibit markedly different interaction-sensitivity characteristics. Trajectron++and LSTM-Interaction show little response to neighboring-agent perturbations, suggesting a strong reliance on the target agent's motion history, whereas AgentFormer is considerably more sensitive to spatial perturbations and FIERY shows substantial degradation in occupancy-based metrics as perturbation magnitude increases. Current architectures thus occupy different points on a sensitivity spectrum ranging from highly insensitive to highly sensitive. This ranking is qualitative rather than strictly quantitative: models are evaluated at different prediction horizons and in different output spaces (Table 3), so absolute sensitivity magnitudes are not directly comparable across architectures. The relative ordering, however, is consistent across all four metrics and both perturbation families evaluated per model, which is the basis for the qualitative ranking.

7.2 Implications and Limitations

Autonomous-driving prediction systems must respond reliably to the behavior of surrounding agents, yet the experiments show that several models remain largely unaffected by realistic neighbor perturbations. Since conventional ADE/FDE metrics cannot reveal this behavior, counterfactual sensitivity analysis offers a complementary diagnostic tool for evaluating interaction awareness before deployment. This study nonetheless evaluates a single checkpoint per architecture, and results may vary across datasets, training runs, or hyperparameter settings, density analysis is reported only for LSTM-Interaction, and perturbations are applied independently rather than in combination. Extending the framework to additional models, datasets, and combined perturbation strategies remains an important direction for future work.

Table 9: Change-magnitude distribution.

Level	%
None (< 0.01m)	65.08
Small (0.01–0.1)	25.75
Med. (0.1–0.5)	5.36
Large (> 0.5)	3.81

Table 10: Error amplification ΔE.

Shift (m)	ΔE
2	-0.000005
5	-0.000011
10	-0.000021
15	-0.000033

8 Conclusion

This paper presented a counterfactual perturbation framework for evaluating the sensitivity of interaction of trajectory prediction models. By applying structural, geometric, and perceptual perturbations to neighboring-agent trajectories, we analyzed how four representative architectures respond to changes in the interaction context. The results show that LSTM-Interaction and Trajectron++ exhibit minimal sensitivity to neighboring-agent perturbations, while AgentFormer and FIERY demonstrate stronger dependence on interaction information. For the LSTM-Interaction, we further observed that interaction sensitivity does not necessarily increase with scene density and may decrease in highly crowded environments (Table 8), whether this pattern holds for architectures with richer interaction mechanisms is an open question we leave to future work (Section 7.2). These findings indicate that conventional accuracy metrics alone are insufficient for understanding how trajectory prediction models utilize neighboring-agent information. The proposed framework provides a complementary evaluation methodology for analyzing interaction behavior in autonomous-driving prediction systems.

AI Use Disclosure

Generative AI tools were used solely for grammar and language editing of the manuscript text (e.g., sentence clarity, phrasing, and copyediting).

References

A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proc. IEEE CVPR*, pp. 961–971, 2016.

B. Bhattacharyya *et al.* MotionLM: Multi-agent motion forecasting as language modeling. In *Proc. NeurIPS*, 2024.

H. Caesar *et al.* nuScenes: A multimodal dataset for autonomous driving. In *Proc. IEEE CVPR*, pp. 11621–11631, 2020.

A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. IEEE CVPR*, pp. 2255–2264, 2018.

R. S. Hallyburton *et al.* Security analysis of camera-LiDAR fusion against black-box attacks on autonomous vehicles. In *Proc. USENIX Security*, 2022.

A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall. FIERY: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proc. IEEE ICCV*, pp. 15273–15282, 2021.

Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang. STGAT: Modeling spatial-temporal interactions for human trajectory prediction. In *Proc. IEEE ICCV*, pp. 6272–6281, 2019.

S. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer. Dynamic environment prediction in urban scenes using recurrent representation learning. In *Proc. IEEE ITSC*, pp. 3477–3483, 2019.

D. Jost, L. Paparusso, M. Stoll, J. Wagner, R. Rajan, and J. Bödecker. Super agents and confounders: Influence of surrounding agents on vehicle trajectory prediction. *arXiv preprint arXiv:2604.03463*, 2026.

D. Kaushik, E. Hovy, and Z. C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *Proc. ICLR*, 2020.

Y. Li, C. Yang, P. Liang, Z. Cao, and Y. Zheng. GRIP: Graph-based interaction-aware trajectory prediction. In *Proc. IEEE ITSC*, pp. 3960–3966, 2019.

P. Liu, H. Liu, X. Liu, Y. Li, J. Chen, Y. He, and J. Ma. Scene-aware explainable multimodal trajectory prediction. *arXiv preprint arXiv:2410.16795*, 2024.

N. Nayakanti, A. Alahi, and S. Savarese. Wayformer: Motion forecasting via simple and efficient attention networks. In *Proc. IEEE ICRA*, 2023.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

M. Pourkeshavarz, J. Zhang, and A. Rasouli. CaDeT: A causal disentanglement approach for robust trajectory prediction in autonomous driving. In *Proc. IEEE/CVF CVPR*, pp. 14874–14884, 2024.

R. Roelofs, L. Sun, B. Caine, K. S. Refaat, B. Sapp, S. Ettinger, and W. Chai. CausalAgents: A robustness benchmark for motion forecasting using causal relationships. *arXiv preprint arXiv:2207.03586*, 2022.

T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proc. ECCV*, 2020.

T. Shi, S. Zhao, K. Fragkiadaki, and H. Zhao. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proc. IEEE ICCV*, pp. 9813–9823, 2021.

S. Shi, L. Jiang, D. Ding, and H. Zhao. UniTraj: A unified framework for scalable trajectory prediction. In *Proc. ECCV*, 2024.

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box. *Harvard Journal of Law & Technology*, 31(2):841–887, 2018.

Y. Zhang, J. Sun, H. Yin, and H. Zhao. On the robustness of 3D object detection in autonomous driving with sensor noise. *arXiv preprint arXiv:2204.07481*, 2022.

Z. Zhou, X. Wang, and Y. Li. QCNet: Query-centric trajectory prediction for autonomous driving. In *Proc. IEEE CVPR*, 2023.

A Evaluation Algorithm

Algorithm 1 summarizes the counterfactual sensitivity evaluation procedure described in Section 3.

Algorithm 1 Counterfactual Sensitivity Evaluation Framework

Input: Dataset $\mathcal{D}$, prediction model f, perturbation set $\mathcal{P}$
Output: S, S_{final}, ZRR, and ΔE
for each sample $(\mathbf{X}_T, \mathbf{X}_N, Y) \in \mathcal{D}$ **do**
    $\hat{Y}_{orig} \leftarrow f(\mathbf{X}_T, \mathbf{X}_N)$
    for each perturbation $p \in \mathcal{P}$ **do**
        $\mathbf{X}'_N \leftarrow P_p(\mathbf{X}_N)$
        $\hat{Y}_{cf} \leftarrow f(\mathbf{X}_T, \mathbf{X}'_N)$
        Compute S, S_{final}, ZRR, and ΔE
    end for
end for
Aggregate metrics across samples
Compute density-wise statistics (low, medium, high)
Report results for each model and perturbation type