

Rashomon in the Streets: Explanation Ambiguity in Scene Understanding

Helge Spieker¹, Jørn Eirik Betten¹, Arnaud Gotlieb¹, Nadjib Lazaar², Nassim Belmecheri¹

¹Simula Research Laboratory, Oslo, Norway

²LISN, Université Paris-Saclay, Saclay, France

{helge,jorneirik,nassim,arnaud}@simula.no, lazaa@lisn.fr

Abstract

Explainable AI (XAI) is essential for validating and trusting models in safety-critical applications like autonomous driving. However, the reliability of XAI is challenged by the Rashomon effect, where multiple, equally accurate models can offer divergent explanations for the same prediction. This paper provides the first empirical quantification of this effect for the task of action prediction in real-world driving scenes. Using Qualitative Explainable Graphs (QXGs) as a symbolic scene representation, we train Rashomon sets of two distinct model classes: interpretable, pair-based gradient boosting models and complex, graph-based Graph Neural Networks (GNNs). Using feature attribution methods, we measure the agreement of explanations both within and between these classes. Our results reveal significant explanation disagreement. Our findings suggest that explanation ambiguity is an inherent property of the problem, not just a modeling artifact.

Introduction

Scene understanding in automated driving is the task of perceiving the vehicle’s environment, including other traffic participants like cyclists and pedestrians, and predicting their actions. This capability is critical for monitoring interactions between traffic participants, anticipating their future behavior, and making informed, safe decisions (Yang et al. 2019; Muhammad et al. 2021, 2022).

Automated driving is a challenging context and its user acceptance requires sufficient trust into the reliability of AI-based systems (Nastjuk et al. 2020; Llorca and Gómez 2021), including the ability to audit and analyze them to understand why certain decisions have been taken (Dwivedi et al. 2023), that is, to introduce a form of model transparency. There are two main approaches towards this transparency: interpretable models and black-box explanations. Interpretable models are white-box and their decision rules can be directly interpreted, such as tree-based models like random forests. Black-box models, including all forms of neural networks, do not allow direct inspections, instead they are explained through additional functionality and techniques from explainable AI (XAI). Interpretable models

have the advantage of not needing external tooling for transparency, which adds overhead and is potentially misleading (Rudin et al. 2022). At the same time, performance-wise neural networks and their variants are the state-of-the-art in many domains.

A challenge inherent to both classes of models is that there can be many well-performing models trained from the same dataset, yet with large differences in their internal structures and encoded rules, nevertheless producing the same predictions. This effect is called the *Rashomon effect* in machine learning (Breiman 2001). It does not only affect model structure and model complexity (Fisher, Rudin, and Dominici 2019), but it has been shown to affect model explanations, too.

Closest to our work, Müller et al. (2023) were the first to highlight the significance of the Rashomon effect for explainable machine learning, demonstrating its effect on several case studies. They train models to give exact same outputs on the dataset, while starting from different initializations, and can still show significant disagreement in the model explanations. Although conducted in a controlled setting, their work highlights the potential pitfalls of the Rashomon effect for the reliability of XAI.

In this work, we investigate the existence of the Rashomon effect for explainable scene understanding in the automated driving domain, trained on a real-world dataset. Starting from symbolic scene graphs in the form of Qualitative Explainable Graphs (QXG) (Belmecheri et al. 2024), we train multiple models that provide good explanations for actions in scene graphs and analyze the alignment of explanations among the models, comparing them to human-labeled scene information. We consider both interpretable random forest models (Belmecheri et al. 2024) and black-box graph neural networks (GNN) (Belmecheri et al. 2025b) as prediction models, covering both perspectives on the transparency challenge. Experiments are performed using the nuScenes dataset (Caesar et al. 2020) with additional labels from DriveLM (Sima et al. 2023).

In summary, we make the following contributions: (1) we show the existence of the Rashomon effect in scene understanding, (2) we analyze explanation agreement within two distinct modeling approaches—one interpretable and one black-box—that use a qualitative scene representation to identify relevant objects, and (3) we contextualize the

observed results for trustworthiness and model inspection, highlighting opportunities for future research.

Background

This section presents the necessary background for our study. It gives an overview and formalization of the Rashomon effect in general and its definition for the context of our work. Additionally, it describes the qualitative explainable graph (QXG), the graph-oriented scene representation used as the data format for the ML models in our experiments.

Rashomon Effect

Introduced and coined by Leo Breiman (Breiman 2001), the *Rashomon effect* refers to a multiplicity of high-performing but *different* models for the same prediction task, i.e., the same train-validation-test split of data. The effect captures the multiplicity of valid *interpretations* of the data, which Breiman highlighted in the varying attribution scores of the input features in his original work.

A *Rashomon set* is a set of models that have been trained using the same training data, with an approximately equally high general performance. Definitions of the Rashomon set are typically tailored to the purpose of the study, resulting in varying definitions. Semenova, Rudin, and Parr (2022) derive mathematical properties from the multiplicity of models able to interpolate a training set. Their definition of the Rashomon set is the set of model architectures capable of fitting the training set within some error, while traditional definitions encompass the set of models performing well on unseen data during training. As highlighted by Breiman, the Rashomon effect has consequences for the *reliance* of feature importance attributions. Fisher, Rudin, and Dominici (2019) propose a model class reliance measure that quantifies the variation in feature importance given a model class. In this work, they present the following definition of the Rashomon set:

Definition 1 (Population ϵ -Rashomon Set). *Given a general data point $(x, y) \sim p_{\mathcal{D}}$, where x is an input drawn from some input space \mathcal{X} , y is an output drawn from some output space \mathcal{Y} and $p_{\mathcal{D}}$ is the true joint population distribution, the Rashomon set \mathcal{R} is a function of the Rashomon parameter $\epsilon > 0$, a loss function $L(f(x), y) \rightarrow \mathbb{R}$, a reference model \tilde{f} , and a model space \mathcal{F} , defined as*

$$\mathcal{R}(\epsilon) = \left\{ f \in \mathcal{F} : \mathbb{E}[L(f(x), y)] \leq \mathbb{E}[L(\tilde{f}(x), y)] + \epsilon \right\}, \quad (1)$$

where the expectation is taken over the population distribution.

Here, ϵ adjusts the maximum difference in loss to account for a model considered to be equally performing, and therefore included in the Rashomon set. The problem with 1 is the intractability of computing the expectation over the true population distribution, which is unknown. A more practical definition will therefore apply a test set, unseen to the model during training, to give an empirical estimate of the population loss.

In this work, we work with the following definition:

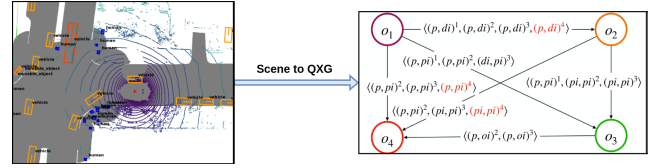


Figure 1: Illustration of the QXG built from a scene.

Definition 2 (Validation ϵ -Rashomon Set). *Given a training set, $\mathcal{D}_{\text{train}} = \{x_j, y_j\}_{j=1}^M$, and a validation set, $\mathcal{D}_{\text{val}} = \{x_i, y_i\}_{i=1}^N$, the Rashomon set \mathcal{R} is a function of the Rashomon parameter $\epsilon > 0$, a loss functional $L(f) \rightarrow \mathbb{R}$, a reference model \tilde{f} , and a model space \mathcal{F} , defined as*

$$\mathcal{R}(\epsilon) = \left\{ f \in \mathcal{F} : \hat{\mathbb{E}}_{\mathcal{D}_{\text{val}}}[L(f)] < \hat{\mathbb{E}}_{\mathcal{D}_{\text{val}}}[L(\tilde{f})] + \epsilon \right\}, \quad (2)$$

where $\hat{\mathbb{E}}_{\mathcal{D}_{\text{val}}}[L(f)]$ denotes the average loss over the validation set, and all models are trained using the same training set, $\mathcal{D}_{\text{train}}$.

Qualitative Explainable Graphs

The Qualitative Explainable Graph (QXG) serves as a scene representation, capturing the qualitative spatial-temporal relationships among objects in a graph structure (Belmecheri et al. 2023, 2024). It has applications to symbolic cause determination (Belmecheri et al. 2025a) and data-driven relevant object identification with graph neural networks (Belmecheri et al. 2025b). For a given scene, denoted as s , the corresponding graph is constructed with a unique node for each object within the set of detected objects in the scene, \mathcal{O} . Edges, contained in the set \mathcal{V} , are established between any two objects that co-appear in at least one frame f .

The labels on these edges are derived from spatial relations that are evaluated for each object pair $(o_i, o_j) \in \mathcal{O}$ at every frame. Formally, this is achieved using a set of algebras $\mathcal{A} = \{A_1, \dots, A_n\}$. Each algebra A_i is defined by a set of relations $R_i = \{r_1, \dots, r_m\}$, with each relation r_i describing a particular spatial property between objects $(o_i, o_j) \in \mathcal{O}$ at a specific frame f_k . Since a scene $s_i \in S$ comprises multiple frames, a new relation r_k is determined from each algebra A_i for every frame f_k , which endows the QXG with spatio-temporal properties. Figure 1 provides an illustration of a QXG built from a three-frame scene; while only the third frame is highlighted, the QXG itself contains information on the dynamics of the spatio-temporal interactions across all three frames.

In this paper, we employ a combination of three qualitative calculi to model the relationships between objects and capture the necessary spatial information. These are: **Qualitative Distance Calculi** (Renz and Nebel 2007) for distance, **Qualitative Trajectory Calculi** (Dylla et al. 2017) for trajectory dynamics, and **Rectangle Algebra** (Renz and Nebel 2007) for relative positioning.

Consider the following example:

Example. *In a given frame f_k , consider two detected objects, o_1 (a car) and o_2 (a pedestrian), with their respective*

bounding boxes $bbox_1$ and $bbox_2$. If we apply the **Qualitative Distance Calculi**, the set of possible relations could be $\{\text{very close, close, far, very far}\}$. The specific relation is determined by computing the Euclidean distance between the centroids of $bbox_1$ and $bbox_2$. The relation is then set according to whether the value of $\text{distance}(bbox_1, bbox_2)$ falls below, exceeds, or lies between thresholds $(\theta_1, \theta_2, \theta_3)$.

However, it is crucial to note that the QXG’s underlying formulation and its application are not dependent on this specific choice of calculi. The framework remains valid as long as the selected calculi possess sufficient expressiveness to describe, at a minimum, the relative positioning and distance of objects. For certain use cases, the representation can be further enriched by integrating additional calculi (Dylla et al. 2017).

Method

In this section, we describe the action explanation problem in scene understanding and describe two approaches for action explanation using qualitative explainable graphs as the scene representation.

Problem Description

Definition 3 (Action Explanation). *Given a scene graph $G = (O, V)$, and an observed action a performed by one of the objects in the scene, identify the object $o \in O$ that has caused the action.*

Put differently, the action explanation problem consists of identifying the relevant object as the cause for an action. The problem is a simplification of the real world in such that actions can be multi-causal or stem from unobserved circumstances, i.e., events within a vehicle or irrational personal decisions. Nevertheless, it captures the essence of the complexity and is suitable for our study, especially since there are no extensive datasets with multi-cause labels that would support studying an extended problem setting.

For the purpose of our study, we consider two action explanation settings that approach the problem from different technical perspectives. The first decomposes the scene graph and uses more interpretable rule-based models, whereas the second works on the full graph but uses more complex graph neural networks.

Pair-based Action Explanation

We follow the indirect approach from Belmecheri et al. (2024) and approach the action explanation problem as a multi-classification problem on all object pairs, consisting of the object having performed the action, e.g., the ego vehicle, and every other object in the scene graph (see Figure 3). This approach first extracts from the overall scene graph a star graph centered around the action-performing object (see Figure 2), which then gets deconstructed into an edge list of all object pairs. Each object pair is classified according to which action it might have caused by a rule-based model, like a decision tree, a random forest, or gradient-boosting decision trees, as in our case. The most relevant object is the object with the highest classification score for the actually observed action.

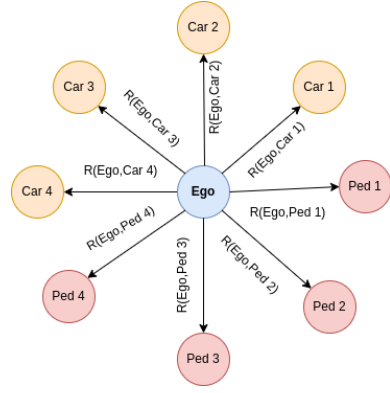


Figure 2: Extracted star graph centered around action-performing object.

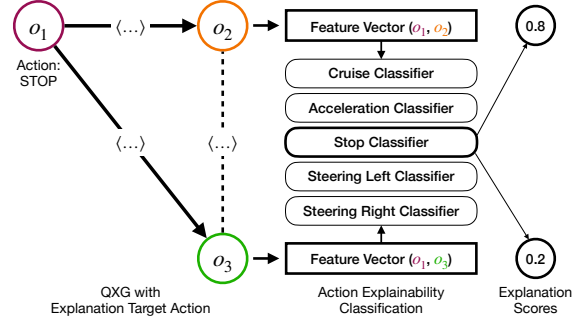


Figure 3: Edge-list-based Action Explanation (adapted from Belmecheri et al. (2024))

Decomposing the graph into an edge list has the advantage that each edge list is of fixed size and can be easily handled by common ML models, like the random forests used in our setting. However, it comes at the cost that there is no shared knowledge about the present objects in the scene, as each object is classified individually and only the explainability scores are aggregated then. Due to the varying number of objects in each scene, this is not trivially realized for models with a fixed input size. To overcome this limitation, other models can be applied, such as the graph neural network discussed in the next section.

Due to the interpretability of the classification model, the rules and reasoning for the selection of the most relevant object having caused the action are accessible, and the overall decision process is transparent to inspection. However, given the size of practical and performant tree-based models, they may require in-depth analysis and alternative explanation techniques, as they are used for full black-box models, to become sufficiently transparent.

Graph-based Action Explanation

We mimic the interpretable action explanation approach, but introduce additional context on the objects in the scene through a graph neural network as the classification model. Specifically, we modify the GNN architecture previously introduced for action-independent relevant object identifica-

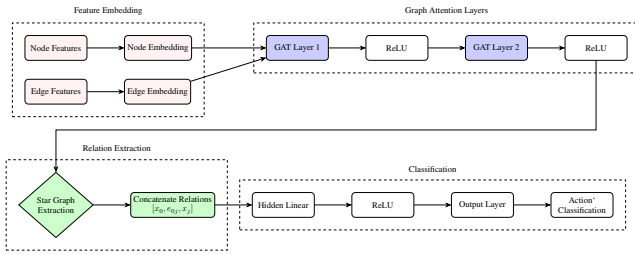


Figure 4: Graph Neural Network architecture for action classification (adapted from Belmecheri et al. (2025b)).

tion (Belmecheri et al. 2025b) (see Figure 4). Whereas the previous architecture performed binary classification of each object’s relevance, our modified architecture is trained on the observed actions and we extract the relevant action-causing objects through explainable ML techniques.

The process works by classifying the scene according to which action is expected to occur. Then the feature attribution according to the true, observed action – which might differ from the predicted action – is calculated to identify the node with the highest attribution, that is, the object deemed most relevant for the observed action. Ways to calculate feature attribution include integrated gradients (Sundararajan, Taly, and Yan 2017) (used in the experiments), Shapley values-based techniques like SHAP (Lundberg and Lee 2017), or other primary attribution technique.

Using the GNN broadens the context of the prediction by incorporating information from each object in the scene, but it comes at the cost of a non-interpretable model and the need for external feature attribution. In the following experiments, we will evaluate whether there is a difference in the ambiguity of the provided explanations within and among the two classes of models.

Experimental Setup

Experiments

We execute two experimental settings, following the two approaches for action explanation, (1) *pair-based action explanation* and (2) *graph-based action explanation*. For each approach, we collect a multiplicity of good models, the validation Rashomon set. We then analyze the existence and prevalence of the Rashomon effect by how well their explanations align with each other.

Dataset

For the experimental validation, we rely on the nuScenes dataset (Caesar et al. 2020) with relevance annotations from DriveLM (Sima et al. 2023). This combination was selected as it is the only available resource that provides labels for object relevance in driving scenarios.

The nuScenes dataset is a large-scale collection of real-world autonomous driving videos, captured across diverse geographic locations and weather conditions. Its comprehensive annotations include 3D bounding boxes for objects such as vehicles, pedestrians, and cyclists in every frame.

Complementing this, the DriveLM dataset from OpenDriveLab is specifically designed to understand driver behavior and object importance. It enhances a subset of scenarios from nuScenes by labeling objects based on their relevance to a driver’s decision-making.

In nuScenes, each video is 20 seconds long and the DriveLM annotations can annotate multiple relevant objects at different times in the video. We generate QXGs of 5 frames for all videos in nuScenes for which there is an actual action of the ego vehicle. The set of actions we consider is the combination of {going straight, steering to the left, steering to the right} \times {driving fast, driving normal, driving slow, not moving}. These actions could be handled separately, but we choose the joint action space to simplify the setup for the experiments and the models. In total, this setup results in 2131 scenes with ego vehicle actions and annotated relevant objects.

Model Selection

We aim to find a validation ϵ -Rashomon Set for the dataset for the interpretable and the black-box action explanation settings. Since both explanation settings are different, we address their training and the selection of the Rashomon set separately rather than as a single population of models.

For the interpretable action explanation, we train 100 gradient boosting decision trees (Friedman 2001) as the classifier with stochastic feature selection and subset sampling (bagging) (Friedman 2002), which avoids overfitting, but introduces randomness into the training process. To select the Rashomon set, we set $\epsilon = 0.05$, meaning we include all models with at least 95% performance of the best model’s performance on the validation set. As the models are very similar in their performance (within 2% performance), this leads to all 100 models being well-performing and being part of the Rashomon set.

For the black-box case, we train 116 GNNs, too, for 200 epochs each. Again, we set $\epsilon = 0.05$, which leads to a selection of 32 well-performing models.

For all models, the train/validation split is fixed, but the randomness in model training, e.g., for data splits and model initialization, is seeded differently per model training.

Data Collection

After the selection of well-performing models, the next step is to collect explanation and measure the models’ agreement. We run all selected models on the scenes with human-annotated relevance labels to explain the action. From the object identified as relevant, we extract the feature importance of the spatio-temporal features encoded in the QXG using SHAP and rank them by their absolute importance. That means, for each of the scenes with an annotated relevant object we run the prediction models, check their correctness, and collect their explanations through an assessment of the feature contributions from SHAP (Lundberg and Lee 2017).

We quantify explanation agreement using two metrics: Fleiss’ kappa and Kendall’s W (described below). We then report aggregated statistics (mean, standard deviation, min, max) for these metrics across all test scenes.

Metrics

The goal of the study is to understand the agreement in perceived feature importance among the models in the Rashomon set.

Fleiss’ Kappa Fleiss’ kappa is a metric to measure the inter-rater agreement among multiple categorizations of items, i.e., how much multiple raters agree in their classification (Fleiss 1971). In our case, we measure how much multiple models agree in their ranking of the most important features for a prediction, i.e. we categorize the features as relevant or not. Fleiss’ kappa is then expressed as the agreement of the top-rated features over a random selection:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \quad (3)$$

In this formula, P_o is the observed proportional agreement among the models, representing the mean agreement calculated across all features. The term P_e is the expected agreement that would occur if the models made their classifications purely by chance, based on the overall distribution of “relevant” and “not relevant” classifications, according to a uniform distribution (Randolph 2005).

The resulting value of kappa indicates the level of agreement, where a value of 1 signifies perfect agreement among all models, a value of 0 suggests the agreement is no better than random chance, and a negative value indicates that the observed agreement is worse than chance, implying systematic disagreement. Generally, higher values of kappa indicate stronger agreement among the models on which features are most important.

For our experiments, we consider a feature “important” if it ranks among the top- k features. We calculate kappa for varying $k \in [1, 20]$, i.e., the top half of features.

Kendall’s W Kendall’s W, also known as Kendall’s Coefficient of Concordance, is a non-parametric statistic used to assess the agreement among several raters (Kendall 1962). It is useful to determine the consensus in ordered data. In our context, it measures how consistently multiple models rank the features based on their importance, moving beyond simple categorization to evaluate the entire ordering.

Kendall’s W is calculated based on the sum of squared deviations of the grand sum of ranks for each feature from the mean. The formula is $W = \frac{12S}{m^2(n^3 - n)}$.

Here, S represents the sum-of-squares statistic over the rank sums, m is the number of raters (our models), and n is the number of items being ranked (the features). Each feature is assigned a rank by each model, and S is computed from the totals of these ranks for each feature.

The value of W ranges from 0 to 1, where a value of 1 indicates perfect agreement, meaning all models produced the exact same ranking of features. A value of 0 signifies a total lack of agreement, suggesting that the rankings provided by the models are essentially random with respect to one another. Unlike correlation coefficients that can be negative (such as Fleiss’ kappa), W does not measure disagreement, only the presence or absence of agreement. Therefore,

a higher value of W signifies a stronger consensus among the models regarding the relative importance of the features.

Technical Setup

Our experiments are implemented in Python, using LightGBM’s gradient boosting decision trees (Ke et al. 2017), scikit-learn’s utilities (Pedregosa et al. 2011), and PyTorch (Ansel et al. 2024) and PyTorch-Geometric (Fey and Lenssen 2019) for the GNN. Explanations are calculated via the Captum library (Kokhlikyan et al. 2020). Experiments are performed on a PC with an Intel Xeon E3-1246 CPU, 24 GB RAM and a NVIDIA GeForce RTX 2060 GPU.¹

Results

We evaluate the models on their predictions on the labeled part of the nuScenes dataset, as described above. Specifically, we analyze their predictions in terms of the agreement on the most influential features, as estimated through SHAP.

Top-K Most Influential Features

In the first analysis, we compare the selection of the most influential, top- k features per scene among the models. Here, we only consider if the same features are part of the top- k most influential features, without considering their order.

We perform the analysis for all $k \in [1, 20]$, which covers the top half of features, and calculate Fleiss’ kappa, i.e., the agreement among the models on which features belong into the top- k . There are 42 features in the pair-based approach and 45 features in the graph-based approach, where the difference stems from a variation in the preprocessing of the rectangle algebra, which gets decomposed into x- and y-positioning in the graph-based approach, and the information on the object types in the pair-based approach, which is part of the node information in the graph-based approach.

The results are shown in Figure 5. Generally, the agreement in the pair-based approach is much higher than in the graph-based approach, except for the first feature. After that, the agreement in the graph-based approach declines, whereas the pair-based approach is steadier. This means that there is more variation in the lower-rank features for the graph-based approach with more fluctuation around the top- k -border, whereas the most influential features are relatively steady among the models with a kappa of 0.6–0.9. This is a reassuring result in that there is an agreement on which features influence the decision the most on what is the relevant object that has caused an action. However, the more features we consider in the ranking, the more disagreement we observe, which is potentially misleading for further analysis of the models, possibly pointing to spurious correlations that are only important for one model, but not for the problem in general. This could indicate that there are *symmetries in the data*, i.e., equivalently relevant features to take a decision if a human were asked, although it is unclear if they are true alternatives or if there are spurious correlations in the training set that are registered by some models.

To better understand the results, we separate the results between all predictions made for an action, including those

¹Online: <https://doi.org/10.5281/zenodo.17045085>

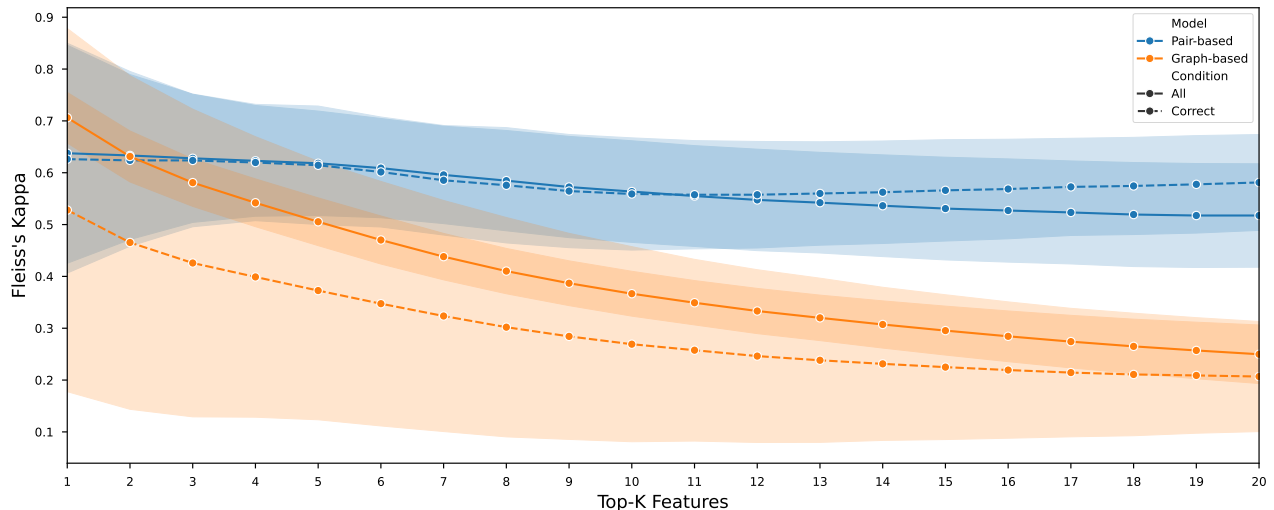


Figure 5: Agreement on top-k features among models. We observe a higher agreement for the Pair-based LightGBM than the Graph-based GNN.

that highlight the wrong object as relevant (respectively not the one that was labeled as relevant), and only those predictions that correctly identify the labeled relevant object. Interestingly, we observe a difference between the two approaches here. In the pair-based approach, the agreement is very similar, independent of the correctness, with a gap for higher top-k values towards a stronger agreement among correct predictions. Wrong predictions might indicate a weakness of that specific model for the given scenario, raising doubts about the reliability of the explanations, whereas correct predictions seem to have a better general understanding of the scene.

Surprisingly, for the GNNs, the agreement among explanations for correct predictions is consistently lower than the agreement across all predictions, albeit with a much larger variance (the shaded area). Again, we might see some symmetries in the data. Otherwise, where the tree-based models of the pair-based approach are constructed according to the training dataset (stopping the depth of additional boosting once the performance plateaus), the capacity of a neural network is set upfront, not entirely independent of the training dataset (as part of the hyperparameter selection), but not coupled to the training process itself. This could be caused by an *overparameterization*, where the model has many redundant pathways to represent the same function, resulting in similar predictions from different internal reasoning.

Feature Ranking Agreement

In the second analysis, we compare the agreement on the ranking of all features. In each analysis, we present the results for both experimental settings jointly, while discussing and highlighting their specifics individually.

To provide a more holistic view of explanation agreement, we use Kendall’s W to measure the consensus on the entire feature ranking, not just the top-k features. The results are summarized in Table 1. The pair-based models show

Model	Condition	Mean	Std. Dev.	Min	Max
Pair-based	All	0.32	0.09	0.08	0.61
Pair-based	Correct	0.40	0.07	0.24	0.61
Graph-based	All	0.07	0.02	0.02	0.24
Graph-based	Correct	0.14	0.18	0.03	1.00

Table 1: Kendall’s W for the feature ranking, separated per explanation approach and prediction correctness. Even though the models are all similarly well-performing, their agreement on feature importance is low (graph-based) to moderate (pair-based) only.

a moderate level of agreement, which improves noticeably when considering only correct predictions. This suggests that when these models correctly identify the relevant object, their underlying reasoning is more consistent.

In contrast, the GNNs exhibit very low agreement on the feature ranking. While the agreement doubles for correct predictions, it remains low overall. The high standard deviation and maximum value of 1.0 for correct GNN predictions indicate that in some specific cases, the GNNs can agree perfectly, but on average, their internal reasoning for the same correct prediction is highly divergent. This reinforces the finding from the top-k analysis: different GNNs in the Rashomon set might learn different, yet equally effective, predictive functions.

Discussion & Limitations

In this section, we interpret our observed results and draw some conclusions on how to embrace or mitigate them. As part of the result discussion, two potential causes for model agreement were raised – *Symmetries in the data* and *Overparameterization* –, which we will discuss specifically. To put the results into context, we discuss the limitations of our ex-

periments and outline paths for further research to advance and broaden them.

Symmetries in the Data An agreement of the models would be easy to interpret, as it confirms with our expectations. Same circumstances should lead to the same conclusions for the same reasons. Disagreements are more difficult. Even if the models draw the correct conclusions, are they doing it for other – equally correct – reasons or for the wrong reasons, i.e., because they were badly trained? For example, there can be multiple causes in a scene to stop the car, albeit the person crossing the crosswalk or the ball rolling on the street. Our experiments cannot answer this questions as we are lacking the necessary data in our dataset. Future work could overcome this by acquiring more data or performing additional studies on varied data, e.g., leaving out objects previously identified as relevant, to understand the relevance of prominent features under data variations.

Overparameterization On the one hand, overparameterization is a generalization of the symmetry issue. A model might cover many ways to explain a prediction, some of which might be correct according to the provided, while there is not sufficient data to exclude the wrong ones. On the other hand, it is a technical problem as the model under training is not suited for the task at hand. It might be too small, relying always on the same few features and not grasping nuances from others. It might be too big, learning too many variations, but not their constraints as there is no limitation, which could potentially be explained by being a form of shortcut learning (Geirhos et al. 2020). Where tree-based model training has its capacity bundled to its performance when using early stopping, this is difficult for black-box models like the used GNNs and it would require a dedicated hyperparameter selection procedure, which is potentially difficult. For the case of automated driving and under consideration of nuScenes and DriveLM, an alternative could be to enforce explanation plausibility into the training paradigm, making a shift from performance optimization, i.e., achieving highest classification scores, towards plausible predictions, i.e., confirming to human expectations. In the case of overparameterization, this is an additional constraint on the final network, reducing the amount of technical symmetries and variations. At the same time, it puts more dependency on the plausibility and reliability of the human annotations – which is a problem itself, especially when considering decisions in automated driving.

Limitations Our study is biased by the focus on the Validation Rashomon set (rather than the focus on the training performance), the choice of ϵ (smaller = tighter and maybe more comparable), that we consider practical models, even though they do not achieve a perfect score (unlike Müller et al. (2023), which explore the extreme point of models with perfect score on their training sets). This approach is more oriented towards a real-world application of the models, where the main selection criterion is their validation score. Potentially, from our definition, we would expect a lower model disagreement than when overfitting the models on the training data, which is an additional study that could

be performed as a conformity check of our results. Finally, our analysis is constrained to one type of interpretable model (gradient boosting) and one GNN architecture. The extent of the Rashomon effect may vary with different model families.

Conclusion

In this paper, we empirically quantified the impact of the Rashomon effect on the stability of explanations for action prediction in autonomous driving. Using a symbolic Qualitative Explainable Graph (QXG) representation, we trained and analyzed Rashomon sets of two distinct model classes: interpretable, pair-based gradient boosting models and complex, graph-based GNNs.

Our findings serve as a cautionary tale against the naive acceptance of post-hoc explanations. To achieve true trustworthiness in AI systems, we must move beyond the illusion of a single “ground-truth” rationale for each prediction. The path forward requires a paradigm shift: instead of asking “Why did this model make this prediction?”, we should ask “What are the possible reasons a good model could have for this prediction?”.

Future work should not seek to eliminate this multiplicity, but rather to understand and leverage it. Promising research directions include: 1) deepening our analysis of the effect, e.g., by addressing inherent explanation sensitivity (Yeh et al. 2019); 2) developing techniques to find a “consensus explanation” that captures the features most consistently identified as important across the Rashomon set; 3) using explanation variance as a new form of uncertainty quantification; and 4) designing training regimes that regularize models to produce more stable explanations, e.g., by using human relevance annotations as an auxiliary loss. By being aware of and embracing the multiplicity of rationales, we can develop a more robust and trustworthy approach to interpreting AI.

Acknowledgments

This work is funded by the AutoCSP project of the Research Council of Norway (grant number 324674) and the European Commission through the projects: AI4CCAM (Trustworthy AI for Connected, Cooperative Automated Mobility, grant agreement No 101076911), MARS (Manufacturing Architecture for Resilience and Sustainability, grant agreement No 101091783), and AI4COPSEC (Boosting EU Copernicus Security and Maritime Monitoring with AI and Machine Learning (ML), grant agreement No 101190021).

References

- Ansel, J.; Yang, E. Z.; He, H.; and et al. 2024. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *ASPLOS*.
- Belmecheri, N.; Gotlieb, A.; Lazaar, N.; and Spieker, H. 2023. Acquiring Qualitative Explainable Graphs for Automated Driving Scene Interpretation. *CoRR*, abs/2308.12755.
- Belmecheri, N.; Gotlieb, A.; Lazaar, N.; and Spieker, H. 2024. Toward Trustworthy Automated Driving through Qualitative Scene Understanding and Explanations. *SAE International Journal of Connected and Automated Vehicles*.

- Belmecheri, N.; Gotlieb, A.; Lazaar, N.; and Spieker, H. 2025a. Automatic Cause Determination in Road Scene Understanding Using Qualitative Reasoning and Four-Valued Logic. In *Advancing Automated Driving in Highly Interactive Scenarios through Behavior Prediction, Trustworthy AI, and Remote Operations @ Intelligent Vehicles (IV)*.
- Belmecheri, N.; Gotlieb, A.; Lazaar, N.; and Spieker, H. 2025b. Explainable Scene Understanding with Qualitative Representations and Graph Neural Networks. In *Advancing Automated Driving in Highly Interactive Scenarios through Behavior Prediction, Trustworthy AI, and Remote Operations @ Intelligent Vehicles (IV)*.
- Breiman, L. 2001. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3): 199–231.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*.
- Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Omer, R.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; and Ranjan, R. 2023. Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Comput. Surv.*, 55(9): 194:1–194:33.
- Dylla, F.; Lee, J. H.; Mossakowski, T.; Schneider, T.; van Delden, A.; van de Ven, J.; and Wolter, D. 2017. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Comput. Surv.*, 50(1).
- Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
- Fisher, A.; Rudin, C.; and Dominici, F. 2019. All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously. *JMLR*, 20: 177.
- Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5): 378.
- Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.
- Friedman, J. H. 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4): 367–378.
- Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*.
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *NeurIPS*, 3146–3154.
- Kendall, M. 1962. *Rank Correlation Methods*. Griffin.
- Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; and Reblitz-Richardson, O. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896.
- Llorca, D. F.; and Gómez, E. 2021. *Trustworthy Autonomous Vehicles: Assessment Criteria for Trustworthy AI in the Autonomous Driving Domain*. Publications Office of the EU.
- Lundberg, S. M.; and Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS*.
- Muhammad, K.; Hussain, T.; Ullah, H.; Ser, J. D.; Rezaei, M.; Kumar, N.; Hijji, M.; Bellavista, P.; and de Albuquerque, V. H. C. 2022. Vision-Based Semantic Segmentation in Scene Understanding for Autonomous Driving: Recent Achievements, Challenges, and Outlooks. *IEEE Trans. Intell. Transp. Syst.*, 23(12): 22694–22715.
- Muhammad, K.; Ullah, A.; Lloret, J.; Ser, J. D.; and de Albuquerque, V. H. C. 2021. Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions. *IEEE Trans. Intell. Transp. Syst.*, 22(7): 4316–4336.
- Müller, S.; Toborek, V.; Beckh, K.; Jakobs, M.; Bauckhage, C.; and Welke, P. 2023. An Empirical Evaluation of the Rashomon Effect in Explainable Machine Learning. In *ECML PKDD*.
- Nastjuk, I.; Herrenkind, B.; Marrone, M.; Brendel, A.; and Kolbe, L. 2020. What drives the acceptance of autonomous driving? An investigation of acceptance factors from an end-user’s perspective. *Technological Forecasting and Social Change*, 161.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; VanderPlas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *JMLR*, 12: 2825–2830.
- Randolph, J. J. 2005. Free-Marginal Multirater Kappa (multirater K [free]): An Alternative to Fleiss’ Fixed-Marginal Multirater Kappa. *Paper presented at the Joensuu Learning and Instruction Symposium (Joensuu, Finland)*.
- Renz, J.; and Nebel, B. 2007. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*.
- Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; and Zhong, C. 2022. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. *Statistics Surveys*, 16(none): 1–85.
- Semenova, L.; Rudin, C.; and Parr, R. 2022. On the Existence of Simpler Machine Learning Models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*.
- Sima, C.; Renz, K.; Chitta, K.; Chen, L.; Zhang, H.; Xie, C.; Luo, P.; Geiger, A.; and Li, H. 2023. DriveLM: Driving with Graph Visual Question Answering. *CoRR*, abs/2312.14150.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *ICML*.
- Yang, S.; Wang, W.; Liu, C.; and Deng, W. 2019. Scene Understanding in Deep Learning-Based End-to-End Controllers for Autonomous Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1): 53–63.
- Yeh, C.-K.; Hsieh, C.-Y.; Suggala, A.; Inouye, D. I.; and Ravikumar, P. K. 2019. On the (In)fidelity and Sensitivity of Explanations. In *NeurIPS*.