RAG-based Explainable Prediction of Road Users Behaviors for Automated Driving Using Knowledge Graphs and Large Language Models

Nazibul Islam Nabil
2222456642

Saif Mohammed
2121913042

Department of Electrical and Computer Science
North South University

nazibul.nabil@northsouth.edu
saif.mohammed@northsouth.edu

October 7, 2025

Publication Information

- **Published Journal:** Expert Systems with Applications
- **Publisher:** Elsevier
- **Published Year:** 2025

- **Authors:** Mohamed Manzour Hussien[*,1], Angie Nataly Melo[1], Augusto Luis Ballardini, Carlota Salinas Maldonado, Rubén Izquierdo, Miguel Ángel Sotelo

- **Keywords:**
  1. Road users' behaviors
  2. Explainable predictions
  3. Pedestrian crossing actions
  4. Lane change maneuvers
  5. Autonomous driving

Abstract

- Predicting road users' behavior is vital for autonomous driving.
- Existing methods rely mainly on kinematic data, ignoring contextual and semantic factors
- Deep Learning models perform well but lack explainability.
- Proposed system combines **Knowledge Graphs (KG)** and **Large Language Models (LLMs)** using RAG.
- Integrates **Knowledge Graph Embeddings (KGE)** and Bayesian inference for inductive reasoning.
- Applied to pedestrian crossing and lane change predictions.
- Achieved **state-f-the-art(SOTA)** performance in anticipation and F1 score.
- Provides **explainable**, **context-aware** predictions for safer autonomous driving.

Introduction

- **Road traffic deaths** remain a major global issue despite safety improvements (WHO, 2023).
- **Vulnerable Road Users (VRUs)** account for 53% of fatalities; pedestrians alone make up 23%.
- **Lane change maneuvers cause** 33% of road crashes (NHTSA, 2023).
- Safety systems must anticipate both **pedestrian** and **driver behavior**.
- Current models rely heavily on **kinematic data**, ignoring**context and social cues**.
- Many AI models lack **interpretability**, acting as ”black boxes”.
- Explainability is key for trust and safety in autonomous vehicles.
- Our system adds **context**, **Bayesian reasoning**, and **RAG-based explanation**.
- Example: system predicts a lane change using TTC and explains it in natural language (Fig 1).
- **Goal:** create explainable, accurate models for safe automated driving.

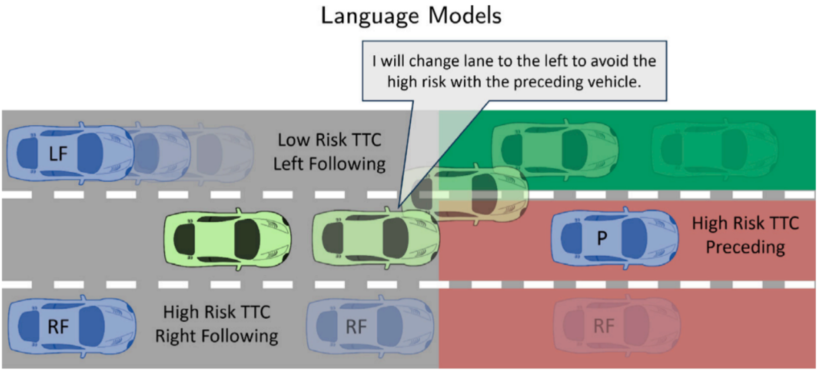

Fig 1. The target vehicle (green) will most likely make a left lane change maneuver based on the risk assessment of the surrounding (blue) vehicles.

Literature Review 1

- **Lane change prediction:**
  - Su et al. (2018) used an **LSTM model** to predict lane changes based on past trajectories and neighboring states.
  - Benterki et al. (2019) **used SVM and ANN models** with velocity, acceleration, lane markings, yaw angle, and yaw rate as inputs.
  - Izquierdo et al. (2019a, 2019b) proposed two methods using the PREVENTION dataset: **MHI-CNN and GoogleNet-LSTM.**
  - Laimona et al. (2020) compared **LSTM and RNN** for lane change intention prediction, concluding **LSTM outperformed RNN** at longer sequences.
  - Xue et al. (2022) applied **XGBoost** for lane change decision prediction and **LSTM** for trajectory prediction using the HighD dataset.
  - Gao et al. (2023) introduced a **dual Transformer model** for lane change and trajectory prediction.

Literature Review 2

- **Pedestrian crossing prediction:** Several approaches include **SingleRNN** (Kotseruba et al., 2020), **CapFormer** (Lorenzo et al., 2021), C3D (Tran et al., 2015),**SFRNN** (Rasouli et al., 2020), and **ConvLSTM** (Shi et al., 2015).
- **Explainability:**
  - Achaji et al. (2023) highlighted **Transformers**' interpretability via attention.
  - Muscholl et al. (2021) proposed a **Bayesian network** incorporating social and interaction cues.

Research Gap

- **Lane change prediction:**
  - Existing behavior prediction models rely mainly on Deep Learning using **kinematic data (speed, position, acceleration).**
  - These models act as **black boxes**, lacking interpretability and explainable reasoning.
  - They ignore **contextual** and **semantic relationships** among road users, environment, and traffic context.
  - Human, social, and environmental factors are rarely integrated into predictions.
  - Current models cannot effectively use prior knowledge or linguistic/contextual information.
  - This limits their ability to generalize in complex, real-world driving scenarios.
  - There is a need for **interpretable models** that combine **contextual understanding**, **reasoning**, and e**xplainable predictions** to build trustworthy autonomous systems.

Research Objective

- Develop an explainable system for predicting road users' behaviors.
- Propose a **neuro-symbolic** framework with **KG, Bayesian inference, and RAG.**
- Build **Knowledge Graphs** encoding linguistic and contextual data.
- Capture relationships among users, environment, and social cues.
- Apply **Bayesian inference** on **KG embeddings** for probabilistic predictions.
- Use **RAG** to generate natural language explanations.
- Ensure transparency, interpretability, and trust in AV decision-making.
- Support validation and standardization of autonomous vehicle systems.

Proposed Method

- The proposed method introduces an **explainable prediction framework** for understanding and forecasting road user behaviors.
- It is composed of three main components:
  1. Knowledge representation of the real world.
  2. Road user behavior prediction approach.
  3. Explainability component.
- The framework integrates **feature extraction and transformation** to build an ontology that captures contextual and behavioral information.
- Two **use cases** are developed:
  1. Pedestrian crossing prediction.
  2. Driver lane-change prediction.
- The following sections detail:
  - Knowledge Graph design for behavior modeling.
  - Prediction methodology using embeddings and Bayesian inference.
  - Explainability through natural language reasoning.

Pedestrian Use Case – Overview and Datasets

- Predict whether a pedestrian will **cross the road within the next 30 frames**.
- Two datasets are used for training and testing:
  - **JAAD (Joint Attention for Autonomous Driving) –** 348 short annotated clips of diverse road scenarios (Rasouli et al., 2017).
  - **PSI (Pedestrian Situated Intent) –** 196 scenes with bounding boxes, textual reasoning, and contextual annotations (Chen et al., 2022).
- Both datasets provide varied pedestrian behaviors under different environments.
- Data imbalance observed: **72 % crossing vs 28 % non-crossing**, causing overfitting toward "crossing".
- Feature extraction and **linguistic transformation** applied to convert numerical data into interpretable linguistic terms.

Pedestrian behavior ontology.

Class	Class description	Instance	Possible relation
Pedestrian	Generic entity pointing to every child pedestrian	Pedestrian	Any
Pedestrian ID	Individual Pedestrian ID	Ped1	HAS_CHILD
Pedestrian instance ID	ID for a pedestrian at a particular frame	Ped1–30	INSTANCE_OF PREVIOUS NEXT
Motion Activity	Pedestrian motion activity	Stand, Walk, Wave, Run, Na	MOTION
Proximity	Pedestrian closeness to the road	NearFromCurb, MiddleDisFromCurb, FarFromCurb	LOCATION
Distance	Pedestrian closeness to the ego-vehicle	TooNearToEgoVeh, NearToEgoVeh, MiddleDisToEgoVeh, FarToEgoVeh TooFarToEgoVeh	EGO_DISTANCE
Orientation	Pedestrian body orientation	VehDirection, LeftDirection, OppositeVehDirection, RigthDirection	ORIENTATION
Gaze	Pedestrian attention	Looking, NotLooking	ATTENTION
Cross Action	Crossing behavior of the pedestrian	crossRoad, noCrossRoad	ACTION

Table-1: Pedestrian behavior ontology

Pedestrian Feature Modeling and Ontology – PedFeatKG

- Extract key pedestrian features and express them as linguistic terms.
- **Motion Activity:** standing, walking, running.
- **Proximity to Road:** near, middle, far.
- **Distance to Ego-Vehicle:** too near, near, medium, far, too far.
- **Body Orientation:** toward, left, right, opposite.
- **Gaze:** looking or not looking at ego-vehicle.
- PedFeatKG ontology encodes features as entities, relations, and instances.
- Pedestrians linked to multiple frames to preserve temporal context.
- Relations (*PREVIOUS*, *NEXT*, *INSTANCE OF*) model continuity.
- Each instance maps to features and action (*crossRoad* / *noCrossRoad*).
- Supports semantic reasoning on features for future action prediction.

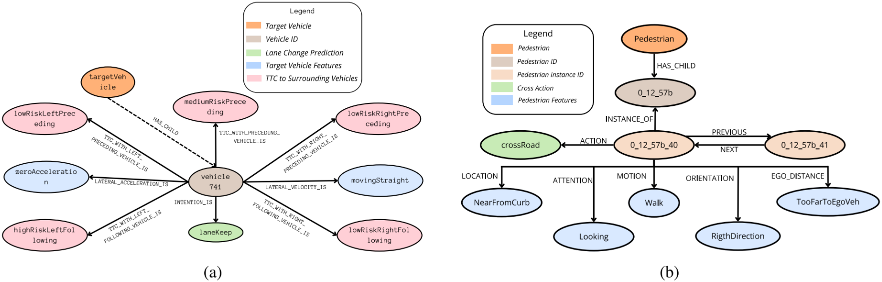

Fig 2. (a) One KG instance where the vehicle has zero lateral acceleration and has medium TTC risk with the preceding vehicle and high TTC with the left following vehicle.(b) PedFeatKG from explainable features with 1 instance.

Vehicle behavior ontology.

Class	Class Description	Instance	Possible Relation
intention	Lane changing intention of the vehicle	LLC (Left Lane Change) LK (Lane Keep) RLC (Right Lane Change)	INTENTION_IS
latVelocity	Vehicle lateral velocity	movingLeft movingStraight movingRight	LATERAL_VELOCITY_IS
latAcceleration	Vehicle lateral accelera- tion	leftAcceleration zeroAcceleration (No lateral acceleration) rightAcceletion	LATERAL_ACCELERATION_IS
ttcPreceding	TTC with the preceding (front) vehicle	highRiskPreceding mediumRiskPreceding lowRiskPreceding	TTC_WITH_PRECEDING_VEHICLE_IS
ttcLeftPreceding	TTC with the left preceding (front) vehicle	highRiskLeftPreceding mediumRiskLeftPreceding lowRiskLeftPreceding	TTC_WITH_LEFT_PRECEDING_VEHICLE_IS
ttcRightPreceding	TTC with the right preceding (front) vehicle	highRiskRightPreceding mediumRiskRightPreceding lowRiskRightPreceding	TTC_WITH_RIGHT_PRECEDING_VEHICLE_IS
ttcLeftFollowing	TTC with the left following (rear) vehicle	highRiskLeftFollowing mediumRiskLeftFollowing lowRiskLeftFollowing	TTC_WITH_LEFT_FOLLOWING_VEHICLE_IS
ttcRightFollowing	TTC with the right following (rear) vehicle	highRiskRightFollowing mediumRiskRightFollowing lowRiskRightFollowing	TTC_WITH_RIGHT_FOLLOWING_VEHICLE_IS
vehicleID	Child vehicle ID which changes every frame	vehicle ID number (e.g. '741')	HAS_CHILD
vehicle	Generic entity pointing to every child vehicle	–	Any

Table-2: Vehicle behavior ontology

Driver Use Case – Lane Change Prediction

- Objective: Predict **lane change maneuvers** on highways.
- Dataset: **HighD** (drone-based German highway recordings).
- Provides precise trajectories, speeds, and vehicle interactions.
- Driver behaviors modeled via feature extraction and linguistic transformation.
- Input features: lateral velocity, acceleration, vehicle intention, and TTC.
- TTC computed with neighboring vehicles (left/right, front/rear).
- Features converted into **linguistic categories** using thresholds.
- Example: Lateral acceleration $\rightarrow$ left/none/right; TTC $\rightarrow$ high/medium/low risk.

Driver Use Case – Lane Change Prediction

- Objective: Predict vehicle **lane changes** on highways.
- Dataset: **HighD**, drone-based German highway recordings.
- Data includes trajectories, speeds, and vehicle interactions.
- Behaviors modeled via feature extraction and linguistic mapping.
- Features: lateral velocity, acceleration, intention, and TTC.
- TTC calculated with neighboring vehicles (front/rear, left/right).
- Numerical features converted into **linguistic categories**.
- Example: Acceleration $\rightarrow$ left/none/right; TTC $\rightarrow$ high/medium/low.

Road Users' Behavior Prediction Approach – Overview

- Both pedestrian and driver prediction use cases follow a common three-phase architecture:
  - **Phase 1:** KG Generation
  - **Phase 2:** Knowledge Graph Embedding (KGE) Learning
  - **Phase 3:** Bayesian Inference and Prediction
- Workflow integrates **feature extraction**, **linguistic transformation**, and **ontology-based reasoning**.
- Combines symbolic knowledge from datasets and expert rules with data-driven embeddings.
- Enables **inductive reasoning** and explainable predictions for both pedestrians and drivers.
- Fig. 3 (in paper) illustrates the end-to-end pipeline connecting KG, KGE, and Bayesian modules.

Method Overview Visual Representation

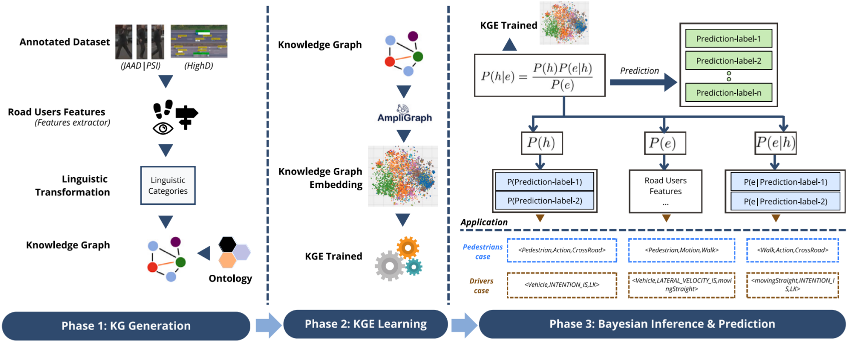

Fig 3. Pipeline architecture for modeling road user's behaviors

Phase 1 – KG Generation Phase 2 – KGE Learning

- **Phase 1 – KG Generation:**
  - Extract behavioral and contextual features from pedestrian (JAAD, PSI) and driver (HighD) datasets.
  - Transform numerical features into **linguistic values** (e.g., near, far, high risk).
  - Construct **Knowledge Graphs (KGs)** as triples ⟨subject, predicate, object⟩ using **AmpliGraph 2.0.0**.
  - Incorporate multiple knowledge sources: dataset annotations, fuzzy rules, and textual explanations.
- **Phase 2 – KGE Learning:**
  - Use **ScoringBasedEmbeddingModel** in AmpliGraph with **TransE** and **ComplEx** scoring layers.
  - Train neural embedding models to represent entities and relations as low-dimensional vectors.
  - Produce **optimal embeddings** encoding contextual and semantic information for inference.

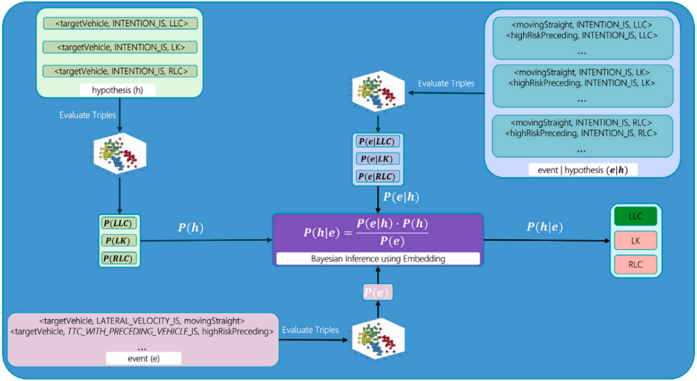

Fig 4. Calculating the lane change probabilities in the Bayesian Inference and Prediction phase.

Phase 3 – Bayesian Inference and Prediction (Part 1)

- Apply **Bayesian reasoning** on learned KG embeddings to make inductive predictions.
- Compute probabilities of reified triples $\langle h, r, t \rangle$ using **AmpliGraph** evaluation scores.
- Combine contextual knowledge and statistical evidence to infer road users' intentions.
- Use **Bayes' Rule:**

$$P(h|e) = \frac{P(h) \times P(e|h)}{P(e)} \tag{1}$$

- Where:
  - h = hypothesis (e.g., crossing / lane change intention).
  - e = evidence (e.g., gaze, proximity, TTC, orientation).
- Integrates **contextual cues** and **linguistic evidence** from the Knowledge Graph.

Phase 3 – Bayesian Inference and Prediction (Part 2)

- Independent evidence probabilities are calculated as:

$$P(e) = P(e_1) \times P(e_2) \times \cdots \times P(e_n) \tag{2}$$

- Conditional evidence given hypothesis:

$$P(e|h) = P(e_1|h) \times P(e_2|h) \times \cdots \times P(e_n|h) \tag{3}$$

- Example – Pedestrian:
  - Evidence: looking at vehicle, near to road.
  - Hypothesis: intention to cross.
  - Compute $P(\text{crossRoad}|e)$ and $P(\text{noCrossRoad}|e)$; choose higher.
- Example – Driver:
  - Evidence: moving straight, high TTC risk with preceding vehicle.
  - Compute $P(\text{LLC}|e)$, $P(\text{RLC}|e)$, $P(\text{LK}|e)$.
- Prediction = label with highest posterior probability.
- Ensures **explainable, probabilistic, and context-aware predictions**.

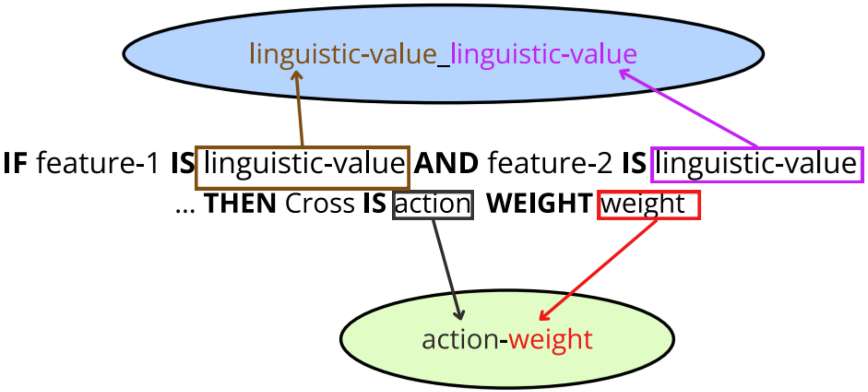

Fig 5. Fuzzy rule conversion definition

Explainability – Fuzzy Rules Approach

- Explainability is achieved through **Fuzzy Logic** and **RAG** methods.
- **Fuzzy logic** mimics human reasoning with multi-valued interpretations.
- Generates **fuzzy rules** linking pedestrian features to crossing actions.
- Uses **IVTURS-FARC** system and **FARC-HD** algorithm for rule mining.
- Fuzzy rules follow the structure:

$$\text{Rule } R_j : \text{ if } x_1 \text{ is } A_{j1},\ x_2 \text{ is } A_{j2},\ \ldots,\ x_n \text{ is } A_{jn} \Rightarrow \text{ Class } = C_j \text{ with weight}$$

- Extracted rules: **60** from PSI and **51** from JAAD datasets.
- Each rule is embedded into the KG as two linked entities.
- One entity connects feature combinations, another links actions with rule weights.
- Ontology **PedFeatRulesKG** extends **PedFeatKG** to include fuzzy rules.
- Provides **interpretable, rule-based explanations** of pedestrian behavior.

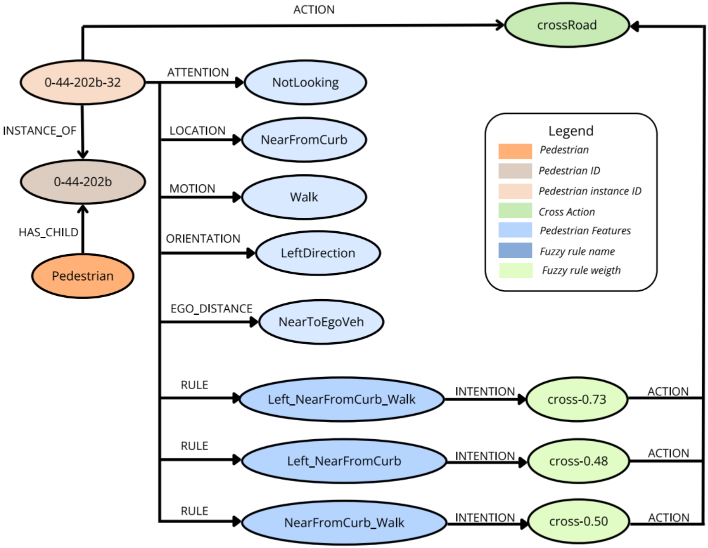

Fig 6. PedFeatRulesKG from explainable features with 1 instance

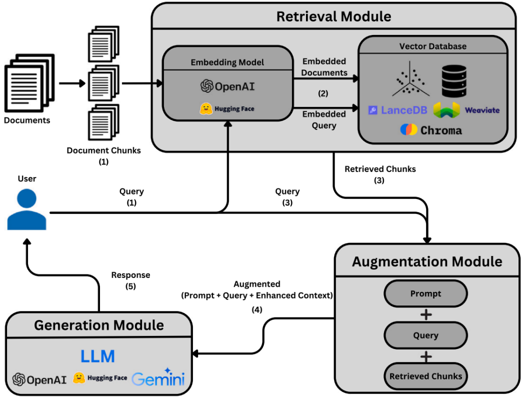

Fig 7. RAG workflow (the numbers show the arrangement of the RAG process flow throughout the figure)

Explainability – Retrieval-Augmented Generation

- **RAG** enhances explainability by combining knowledge retrieval with language generation.
- Integrates pre-trained **LLMs** with an external **retrieval database**.
- The database can be **public or domain-specific**, built from structured or textual data.
- RAG consists of three modules forming its processing pipeline.
- **Retrieval Module:** embeds and stores document vectors (e.g., Chroma, LlamaIndex).
- **Augmentation Module:** retrieves the most relevant chunks for each query.
- **Generation Module:** LLM generates clear, context-based explanations.
- Similarity retrieval uses metrics such as *cosine similarity*.
- Applied to both **pedestrian crossing** and **vehicle lane-change** prediction use cases.
- Produces **human-understandable, transparent justifications** for model predictions.

Implementation and Experimental Results – Overview

- Describes implementation and experiments for both **pedestrian** and **driver** behavior prediction.
- Both use cases share the same **KGE learning** and **Bayesian inference** schemes.
- Implementation highlights:
  - Data preprocessing and feature extraction.
  - Knowledge Graph (KG) generation and embedding.
  - Explainability via RAG (Retrieval-Augmented Generation).
- Performance evaluated using **Precision, Recall, F1-score, and Accuracy**.
- Results compared against several **state-of-the-art methods**.

Pedestrian Features Extraction.

Feature	Extraction type	Description
Motion activity	Neural network	We implemented a transformer architecture that processes the 2D body pose and outputs the pedestrian action.
Proximity to the road	Neural network and estimation	From YOLOPv2 (Han, Zhao, Zhang, Chen, Zhang, & Yuan, 2022) was obtained from the drivable road area segmentation and lane detection. Based on an experimental minimum distance it is estimated whether the pedestrian is near to the road or not.
Distance	Estimation	Estimated using the triangle similarity
Orientation	Neural network	Using the PedRecNet (Burgermeister & Curio, 2022) the joint positions of the human body and the body orientation from the azimuthal angle φ were obtained.
Gaze	Estimation	We used the 2D body pose detection and the positions of the nose, left eye, and right eye keypoints.

Table-3: Pedestrian Features Extraction

Dataset	Ontology	Triples
PSI	PedFeatKG	238 795
	PedFeatRulesKG	302 574
JAAD	PedFeatKG	139 624
	PedFeatRulesKG	197 381

Table-4: Number of triples in the experimental setup

Implementation Details – Pedestrian Use Case

- Implementation includes three stages: feature extraction, KG modeling, and explainability.
- **Feature Extraction:** neural networks implemented with Python and PyTorch.
- **KG Modeling:** built using TensorFlow and AmpliGraph with the ComplEx model.
- **Explainability:** achieved through a LangChain-based RAG framework.
- **Hyperparameters:** embedding size 150, learning rate 0.0005, batch size 10k, corruption 5–20.
- **Evaluation:** Mean Reciprocal Rank (MRR) metric with early stopping.
- **Ontologies:** PedFeatKG and PedFeatRulesKG applied for experiments.
- **RAG Setup:** features converted to text, embedded with OpenAI, stored in Chroma DB.
- Explanations generated using **GPT-4 LLM** from textual context.
- **Data Split:** JAAD (136/35) and PSI (104/48) for training and testing.

Experimental Results (Pedestrian)

Table 5

Comparing the pedestrian behavior predictor with various methods. The table includes the available results.

(a) $JAAD_{test}$

Model	F1	Precision	Recall	Accuracy
C3D	0.65	0.57	0.75	0.84
PCPA	0.68	–	–	**0.85**
Decision Tree	0.78	0.78	0.78	0.78
Fuzzy Logic	0.75	0.69	0.81	0.69
PedFeatKG	0.86	0.77	**0.96**	0.79
PedFeatRulesKG	**0.87**	**0.86**	0.88	0.83

(b) PSI_{test}

Model	F1	Precision	Recall	Accuracy
eP2P	0.66	–	–	0.76
Ours Black Box	0.75	0.74	0.75	0.62
Decision Tree	0.63	0.63	0.63	0.63
Fuzzy Logic	0.72	0.74	0.70	0.59
PedFeatKG	0.81	**0.75**	0.89	0.69
PedFeatRulesKG	**0.84**	**0.75**	**0.94**	**0.72**

Table-5: Comparison of pedestrian behavior predictors with various methods

Experimental Results (Driver)

Table 6

Our Proposal				Decision Trees
1 Second	Precision (%)	Recall (%)	F1-score (%)	F1-score (%)
LK	98.33	96.96	97.64	97.69
LLC	97.98	97.50	97.74	98.03
RLC	97.00	99.42	98.19	97.91
Macro avg	97.77	97.96	**97.86**	**98.88**
2 Seconds	Precision (%)	Recall (%)	F1-score (%)	F1-score (%)
LK	98.86	96.96	97.95	97.19
LLC	97.50	99.15	98.32	97.36
RLC	96.52	98.66	97.58	97.04
Macro avg	97.66	98.25	**97.95**	**97.20**
3 Seconds	Precision (%)	Recall (%)	F1-score (%)	F1-score (%)
LK	92.53	96.96	94.70	93.19
LLC	95.71	91.77	93.70	90.38
RLC	95.46	89.50	92.38	91.71
Macro avg	94.56	92.74	**93.60**	**91.76**
4 Seconds	Precision (%)	Recall (%)	F1-score (%)	F1-score (%)
LK	69.63	96.96	81.05	79.04
LLC	91.30	46.00	61.16	46.19
RLC	88.75	42.39	57.37	53.61
Macro avg	83.22	61.78	**66.52**	**59.61**

Table-6: Precision, recall and F1-score metrics of predictions from our proposed model at different instants

Pedestrian Use Case

Prompt

You are an expert on the pedestrian behavior on the road and you are an expert assistant for question-answering tasks.

- Context (input): Natural text that describes the scene. Linguistic inputs describe the pedestrian state: orientation, gaze, action, proximity to the road and distance.

- Output: Justification of the pedestrian crossing behavior in clear and precise English with a maximum of 75 words and do not use conditional words as may

- Command: Use the following pieces of retrieved context to answer the question. If you don't know the answer, just say that you don't know.

- Example:
Input: Explain the reason why a pedestrian will not cross the street, given the following prior information:

- The pedestrian is looking at the driver for eye contact
- The pedestrian is standing in a leftward direction
- The pedestrian is located at a near distance from the curb
- The pedestrian is far from the vehicle

Output: The pedestrian will not cross the street because they are seeking eye contact with the driver, indicating a need for assurance of safety before crossing. Standing in a leftward direction suggests they are not yet ready to cross, possibly waiting for the right moment or assessing the vehicle's distance and speed. Being far from the vehicle, they might judge it unsafe or premature to cross due to the distance, preferring to wait until the vehicle is closer or has passed.

Query

Explain the reason why a pedestrian will cross the street, given the following prior information:

- The pedestrian is looking at the driver for eye contact.
- The pedestrian is walking quickly in a leftward direction.
- The pedestrian is located at a near distance from the curb.
- The pedestrian is far from the vehicle

Drivers Use Case

Prompt

You are an expert vehicle lane change prediction tasks and can justify why the vehicle did one of the following maneuvers (left lane change, or right lane change, or kept lane).

- Context (input): Natural text that describes the scene. The given inputs are: the target vehicle's lateral acceleration, lateral velocity, TTC (Time To Collision) risk with (left preceding, preceding, right preceding, left following, right following) surrounding vehicles, and the target vehicle's predicted lane change maneuver (high TTC is the highest risk level, medium TTC is risky as well but is less than high, low TTC indicates no risk).

- Output: Justification of the lane change maneuver in clear and precise English with a maximum of 75 words.

- Command: State the main reasons contributing to the anticipated vehicle maneuver. Take the retrieved scenarios as supporting scenes. Follow the next example while generating your output.

- Example:
Input: Explain the reason why the vehicle will make a left lane change, given the following prior information:

- The vehicle has zero lateral acceleration.
- The vehicle is moving straight.
- The TTC risk from the left preceding vehicle is low.
- The TTC risk from the preceding vehicle is medium.
- The TTC risk from the right preceding vehicle is low.
- The TTC risk from the left following vehicle is low.
- The TTC risk from the right following vehicle is low.

Output: (The vehicle will initiate a left lane change in response to the small gap with the vehicle directly ahead, which elevates the risk of the current situation. However, the left lane presents a safer option because the Time-to-Collision (TTC) risk associated with both the preceding and following vehicles in that lane is low. This provides a more secure gap for the maneuver. Consequently, transitioning to the left lane is the best option.)

Query

Explain the reason why the vehicle will keep lane, given the following prior information:

- The vehicle is not accelerating in the lateral direction.
- The vehicle is moving straight.
- The TTC risk from the left preceding vehicle is low.
- The TTC risk from the preceding vehicle is medium.
- The TTC risk from the right preceding vehicle is low.
- The TTC risk from the left following vehicle is high.
- The TTC risk from the right following vehicle is low.

Fig 8. RAG prompt example for road users behavior

Implementation Details – Driver Use Case

- Dataset: **HighD** drone-based dataset, divided by tracks (80
- Trained KG embeddings using **AmpliGraph**.
- Compared two scoring models:
  - **TransE** and **ComplEx**.
- **Training parameters:** embedding size = 100, learning rate = 0.0005, batch size = 10,000, 5 negative triples per positive, SelfAdversarialLoss.
- Early stopping with MRR metric and patience of 5 epochs.
- Hardware: Lenovo Legion i9-13900HX, 32 GB RAM, RTX 4090 GPU.
- **RAG Setup:**
  - Data chunk size = 384 tokens.
  - Embeddings via all-MiniLM-L6-v2 (Hugging Face).
  - Vector DB: Chroma; Generator: OpenAI GPT-4.

Results – Pedestrian Behavior Prediction

- Compared with state-of-the-art methods: C3D, PCPA, Decision Tree, Fuzzy Logic, and eP2P.
- **Our Models:** PedFeatKG and PedFeatRulesKG.
- **Results (JAAD dataset):**
  - PedFeatRulesKG: F1 = 0.87 (↑22
  - PedFeatKG: F1 = 0.86.
- **Results (PSI dataset):**
  - PedFeatRulesKG: F1 = 0.84 (↑18
- **Observations:**
  - Fuzzy rules strengthen the KG's reasoning ability.
  - Integration of multiple knowledge sources enhances robustness.
  - Slightly lower accuracy vs "black box" models, but far higher explainability.

Results – Driver Lane Change Prediction

- Dataset: HighD; tested on 12 tracks (20% test set).
- **TransE model outperformed ComplEx:** F1 = 93.6% at 3 seconds before lane change.
- **Performance:**
  - Maintains F1 more than 90% up to 3 seconds before maneuver.
  - Online inference speed: 3 FPS (330 ms/prediction).
  - Offline speed: up to 500 FPS (2 ms/prediction).
- **Comparison with prior works:**
  - Outperformed Xue et al. (2022) and Gao et al. (2023) beyond 1.5 s horizon.
  - Comparable F1 at short-term (0.5–1.0 s).
- Demonstrates the benefit of **linguistic + semantic reasoning** over pure numerical inputs.

Explainability Examples
Pedestrian Case

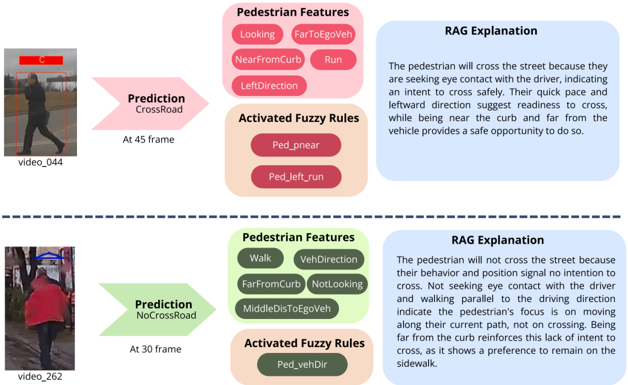

Fig 9. Examples of prediction explainability from JAAD dataset

a) Frame capture at t = -6 s (no action yet).

b) Frame capture at t = -4 s (LLC prediction).

c) Frame capture at t = -2 s (vehicle is merging).

d) Frame capture at t = 0.5 s (crossed lane marking).

Fig 10. Scene explanation through four different frames

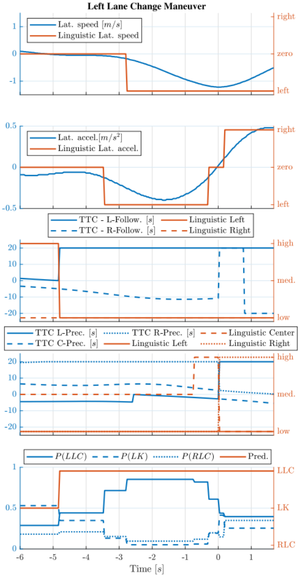

Fig 11. Temporal sequence of numerical variables and linguistic categories

Explainability Results – Pedestrian and Driver Use Cases

- **Pedestrian Use Case:**
  - PedFeatKG provides feature-based reasoning.
  - PedFeatRulesKG links features with fuzzy rules for richer explanations.
  - RAG generates natural-language justifications.
  - Example (JAAD video): *"Pedestrian will cross because they are near the road and oriented left."*
- **Driver Use Case:**
  - Model explains decisions across frames (6s $\rightarrow$ 0.5s before lane change).
  - Bayesian probabilities track risk evolution and lane-change likelihood.
  - RAG explanations correspond to scenario states (Fig. 12).
  - Computation time for explanation 2–3 seconds/query.
- Results highlight strong **interpretability, reasoning transparency, and real-time feasibility**.

Table 7

Comparison with other models using the F1-score (%) metric.

Algorithm \ Prediction Time	0.5s	1.0s	1.5s	2.0s
Xue et al.	98.20	97.10	96.61	95.19
Gao et al.	**99.18**	**98.98**	97.56	91.76
Ours	97.72	97.86	**98.11**	**97.95**

Table-7: Comparison with other models using the F1-score (%) metrics

Use Case	Link
Pedestrians	https://www.youtube.com/playlist?list=PLAeK3AuwxenEqDvdJAk8X9Ysn5egmGvKO
Drivers	https://www.youtube.com/playlist?list=PLAeK3AuwxenFsZslUIYk1CitWKAeAddgt

Table-8: Different multimedia for results visualization in the pedestrians and drivers use cases

Explainability Examples
Driver Case

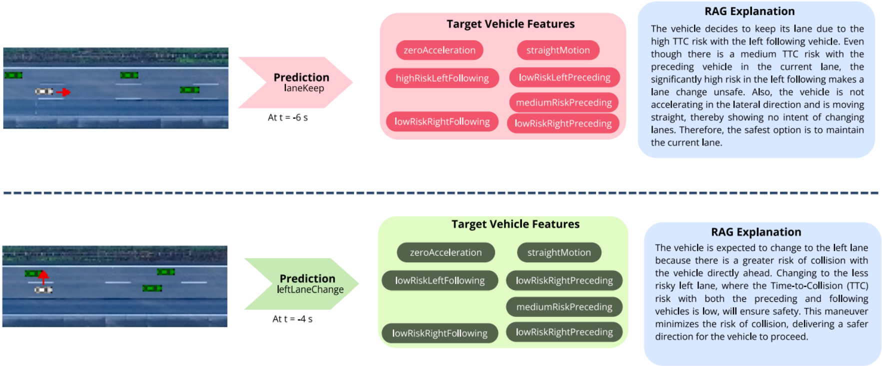

Fig 12. Examples of lane change prediction explainability based on the discussed scene from HighD dataset

Conclusions

- Developed a **context-based behavior prediction system** using Knowledge Graphs and Bayesian inference.
- **Knowledge Graphs (KGs)** enable semantic knowledge representation.
- **Bayesian inference** supports inductive, probabilistic reasoning.
- Implemented two use cases: **pedestrian crossing** and **vehicle lane-change** prediction.
- Outperformed state-of-the-art models in both **F1-score** and **anticipation accuracy**.
- Achieved strong performance even with limited kinematic information.
- Integrated numerical and linguistic data into a unified KG-based framework.
- Incorporated **human knowledge** through linguistic descriptions and rule-based reasoning.
- **Explainability** achieved via RAG combining KG reasoning and LLM expressiveness.

Future Work and Outlook

- Extend predictive capabilities to new use cases:
  - **Near-miss or crash lane-change maneuvers.**
  - **Occluded pedestrians** in urban environments.
- Expand research toward a more **holistic and cross-cultural** understanding of road user behavior.
- Collect new data from diverse regions:
  - **MENA**, **Southeast Asia**, and **Latin America**.
- Integrate the proposed system with the **behavior planner of Autonomous Vehicles (AVs)** to make AV decisions more **human-like and context-aware**.
- Continue exploring hybrid neuro-symbolic architectures for **interpretable AI in autonomous systems**.
- Support from the **HEIDI Project (EU Grant 101069538)** acknowledged.