06/11/2025

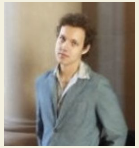
Uncertainty & Safety

In Autonomous Driving

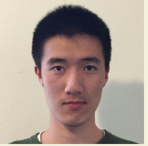
Aleksandr Petiushko (a.petiushko@gmail.com)
Head of AI Research @ Elea
Head of ML Research @ Nuro (formerly)

Uncertainty Quantification for Computer Vision @ CVPR-2025 Workshop

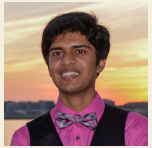
The Team



Vladislav Isenbaev



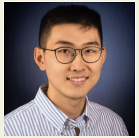
Zhenli Zhang



Shashank Ojha

Constantin Hubmann

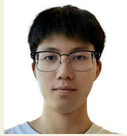
Nima Mohajerin



Yu Yao

Taiqi Wang

Jiawei Zhang



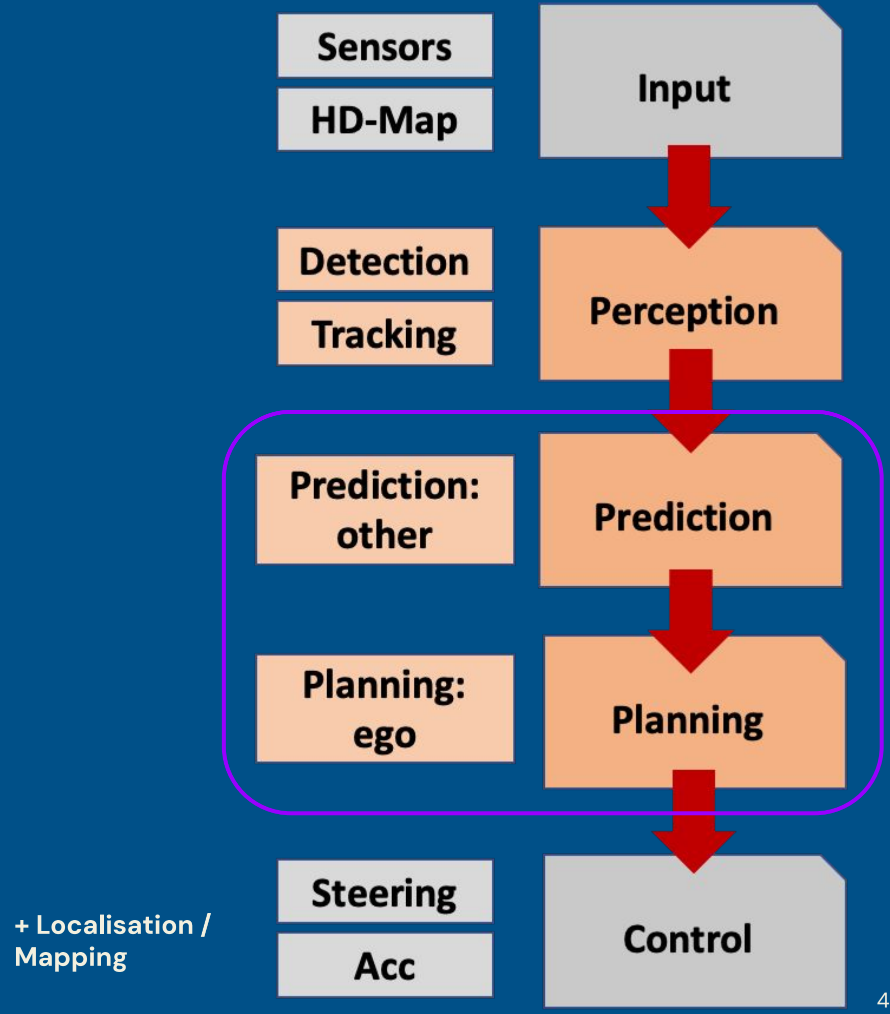
Xuan Yang

Bo Li

1. Uncertainty and Safety
2. Uncertainty in AD: Prediction
3. Safety in AD: Planning
4. Conclusion

AD Stack

- The simplified overview of the *classical* Autonomous Driving (AD) Stack
  - Let's focus on *Prediction* and *Planning* in AD



Uncertainty

- *Prediction* of other agents' motion
- Lack of training data – **Epistemic Uncertainty**
  - It *can* notify the problematic events

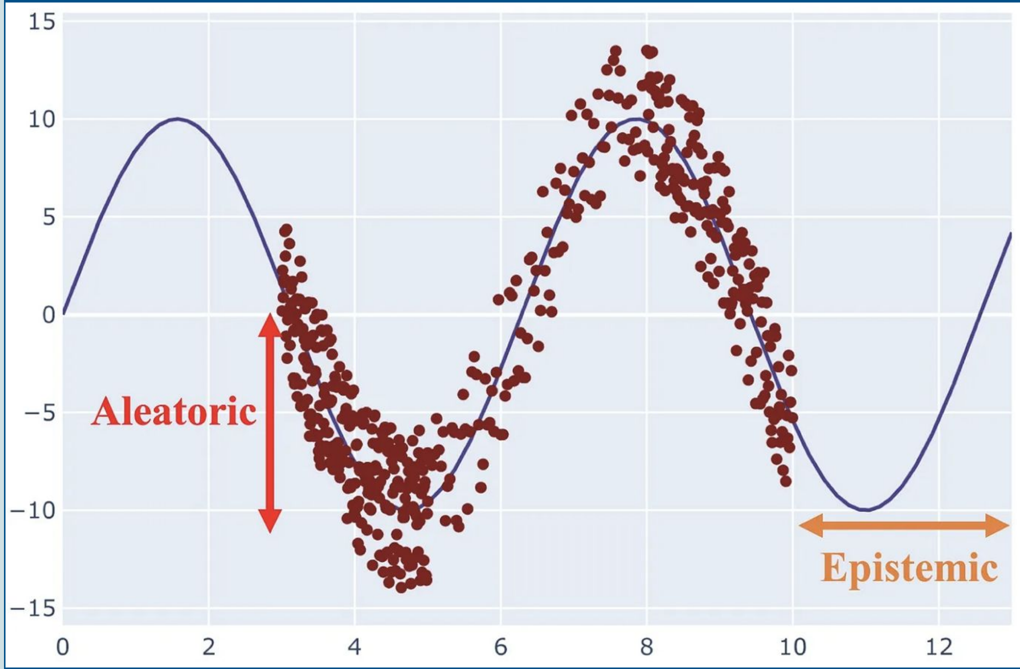


Image source

Safety

- What about safe *Planning*?
- Let's use the **Markov Logic Network** [1] !
    - California Driver's Handbook

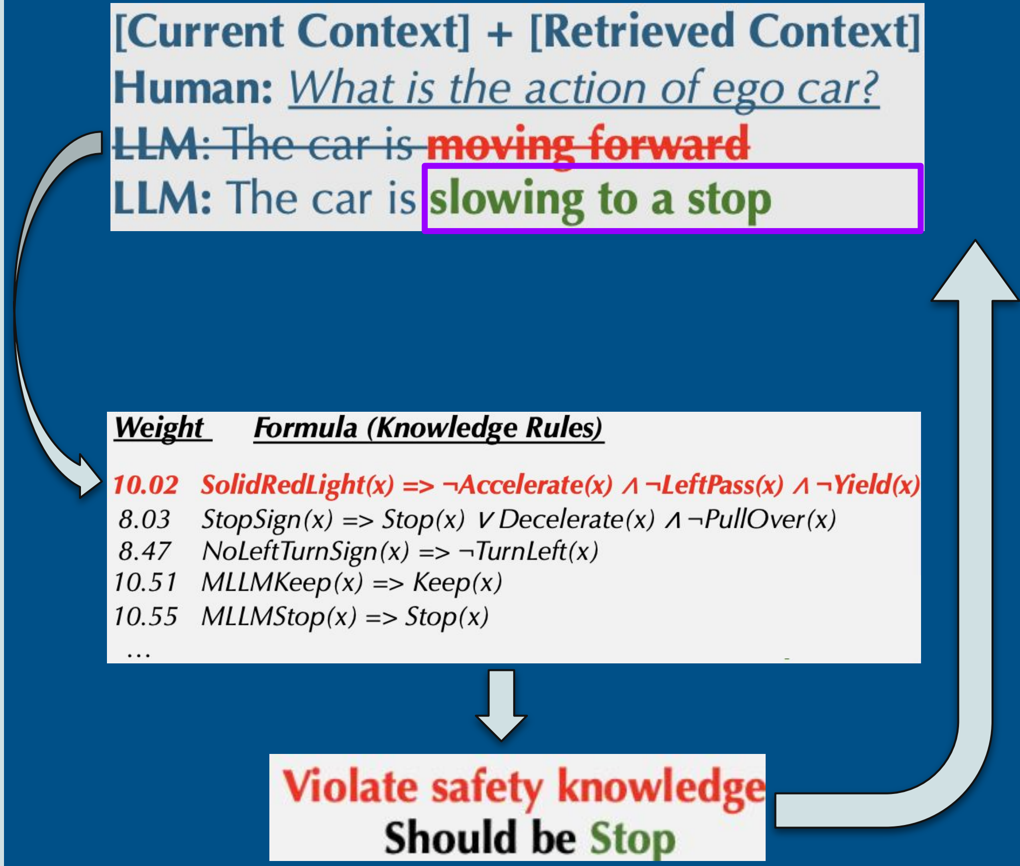


Image source

[1] Markov Logic Networks

1. Uncertainty and Safety
2. **Uncertainty in AD: Prediction**
3. Safety in AD: Planning
4. Conclusion

Uncertainty: Motivation

- Previous version used a highly SW-optimized version of a **Bayesian** Filter
  - A combination of a ***Predictor*** (what to *imagine*) and a ***Tracker*** (what we *see*)
- Unfortunately, it highly depends on the *Predictor* model
- Goals:
  - To design the approach **mostly independent on** the specific model **architecture**
  - To deal with problems of **training data incompleteness**

$$p_t = p_{t-1} \cdot P(z_t|Predictor)+$$

$$+(1 - p_{t-1}) \cdot P(z_t|Tracker)$$

where:

- p_t ***and*** p_{t-1}: current and previous timestamp probabilities of the trajectory
- z_t: trajectory

Uncertainty: Idea

"***Out of Distribution***" (OOD) concept

Idea:

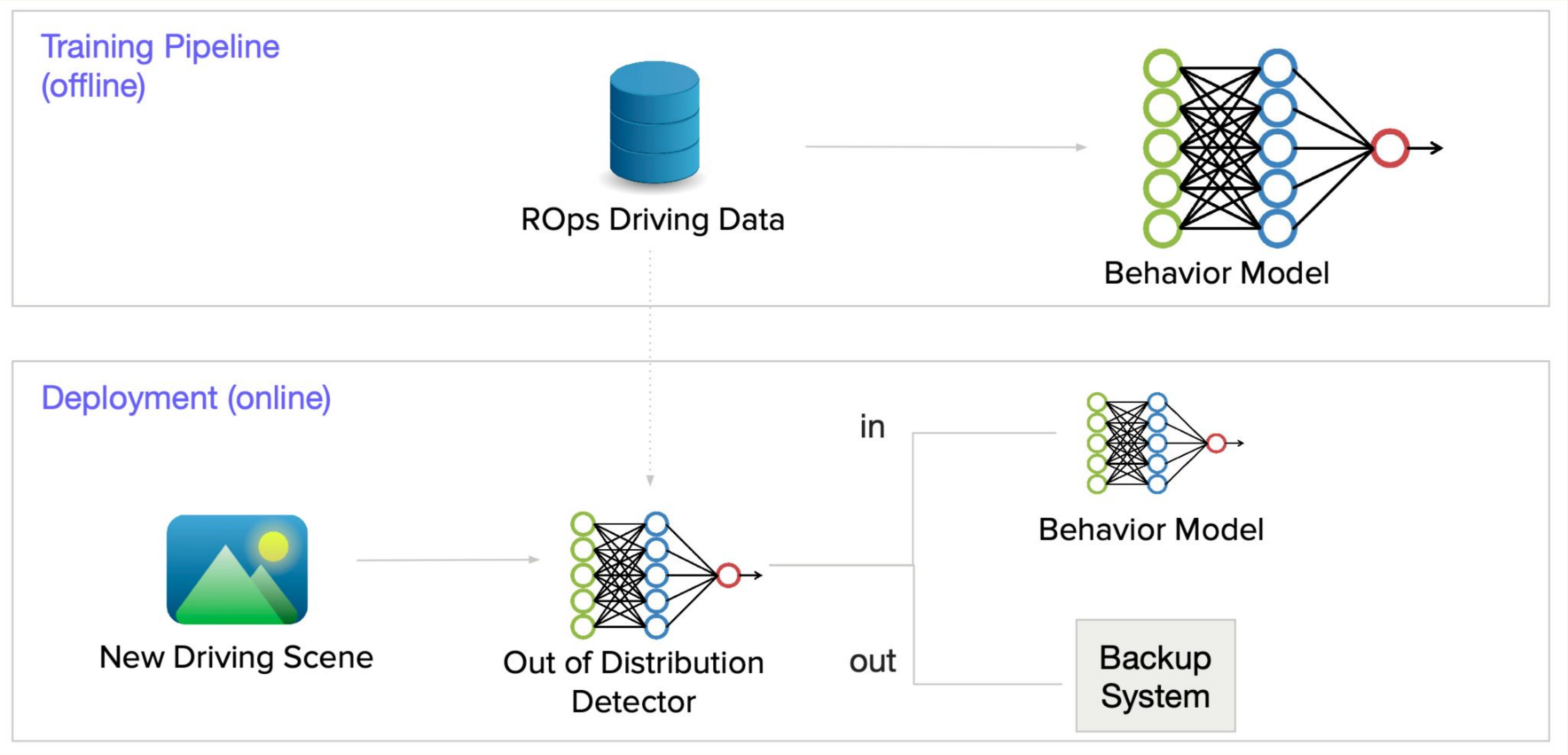
- Measure the OOD at the input of any ML model
- OOD = "*haven't seen similar data during training*"
- It is **Epistemic** (input) **Uncertainty** (lack of data)

Core **assumption**:

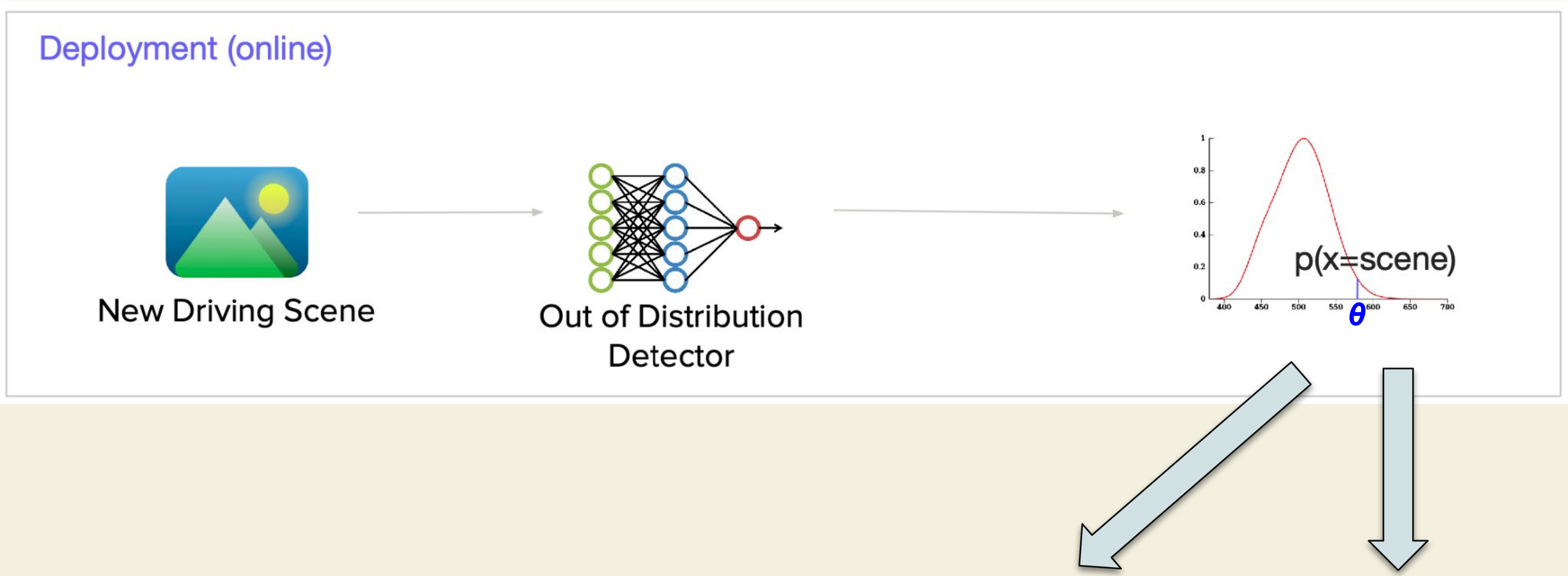
- We cannot guarantee anything about model's output if the input is OOD: it can be either **good** or **bad**

Image source

Implementation Scheme



OOD Module



θ > *th* : OOD! θ <= *th* : common

Uncertainty: Approach

Proof-of-Concept approach:

- **Density Estimator** (DE) on top of the inputs to the Predictor
  - DE will provide the probability (or, equivalently, *NLL*) of the input to be **in distribution** (ID)
- **Training dataset** for DE = (subset of) Predictor training dataset
- **Inputs** = outputs of Behavior Encoder
- Levels of **granularity**:
  - *Scene* level at a specific *timestamp*
  - Even (*track_id, ts*) level

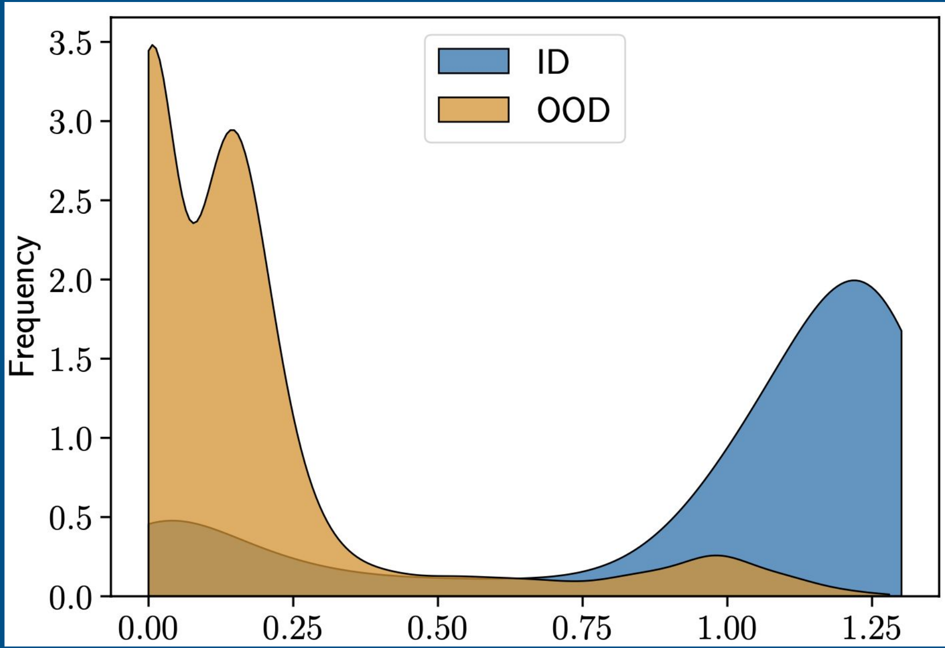


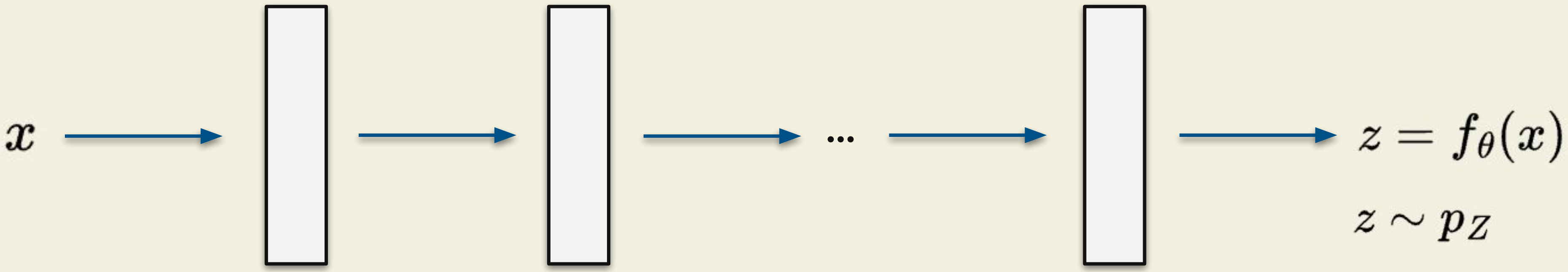
Image source

Uncertainty: First Tries

We started with **Masked Autoregressive Flow** [1] method

Idea: to use the invertible transformations and **map the input** to a **known distribution** (e.g., *Gaussian*)

(initially did some experiments with *different* embedding *sizes*: full and flatten)



[1] Masked Autoregressive Flow for Density Estimation

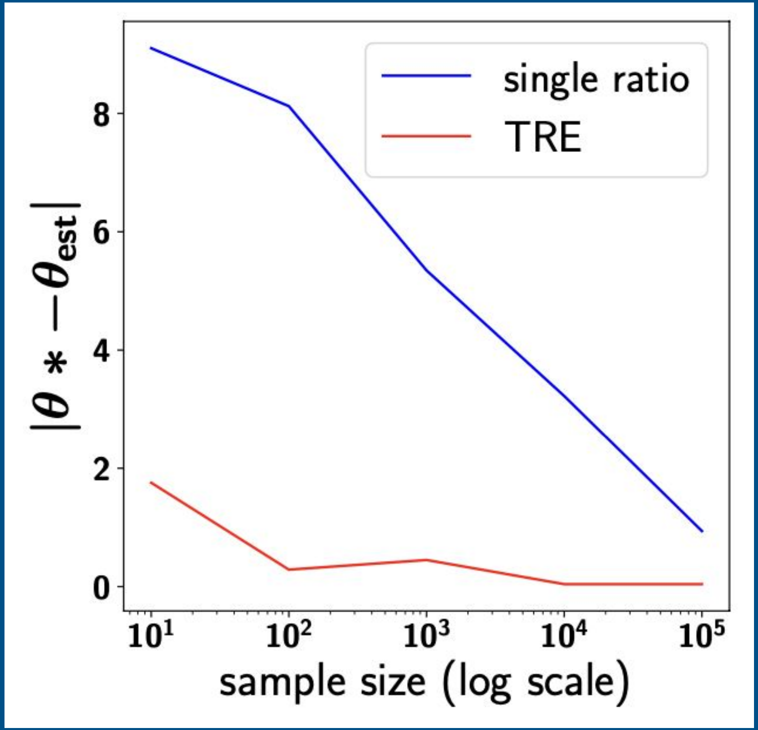
Uncertainty: Final Try

Later, we continued with **Telescoping** Density-**Ratio Estimation** (TRE [1])

Idea:
- Add multiple levels of *noise* to the input and *classify* these levels of noise
- Make multiple intermediate steps
  - $x_k = \sqrt{1-\alpha_k^2} \cdot x_0 + \alpha_k \cdot x_m$
- *Loss* – Multinomial CE
  - π_i – prior class probability
  - h_i – unnormalized logits

$$L(h_1, \ldots, h_C) = \sum_{c=1}^{C} \pi_c \mathbb{E}_{x \sim p_c} \left[-\log \pi_c - h_c + \log \sum_{k=1}^{C} \pi_k \exp\left(h_k(x)\right) \right]$$

[1] Telescoping Density-Ratio Estimation



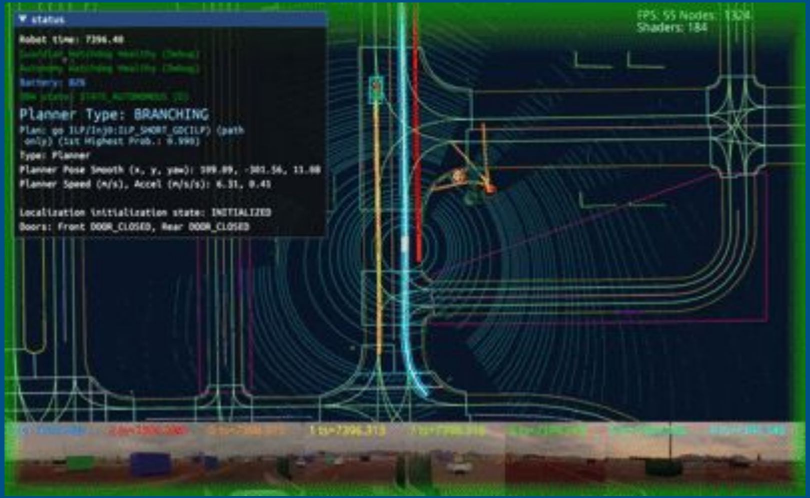
Uncertainty: Eval Preparation

Datasets:

- **OOD**: a number of scenes from the Predictor Eval with the **high** Average Displacement Error (**ADE**) further filtered out by human experts
- **Lowest ADE** scenes
- **Train** scenes - a subsample of the Predictor Eval

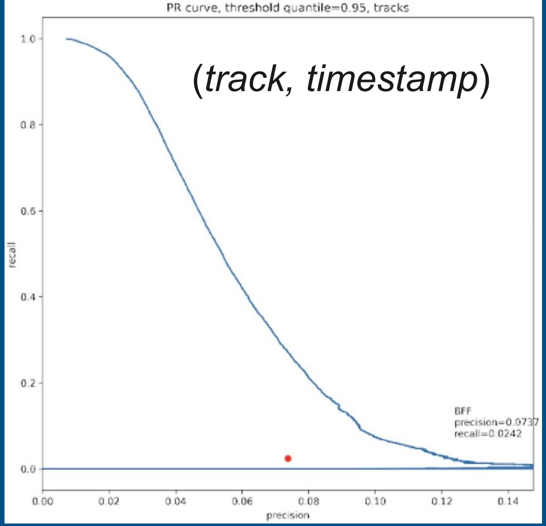
Granularity:

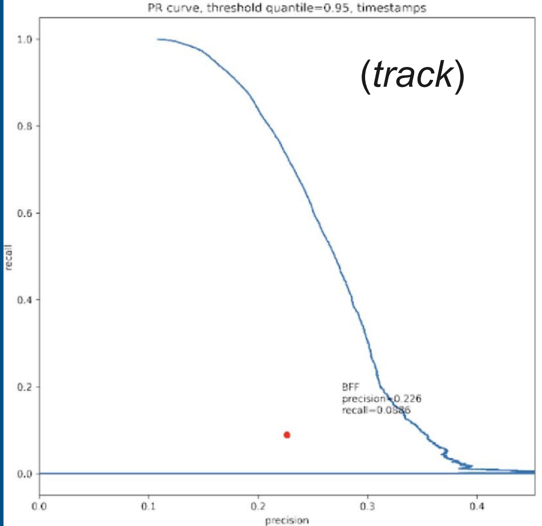
- A positive example at the specific ts if having a high DE
- Everything else is a negative one



Uncertainty: Precision vs Recall

- The closer to the ***upper right*** (higher precision and recall), the ***better***

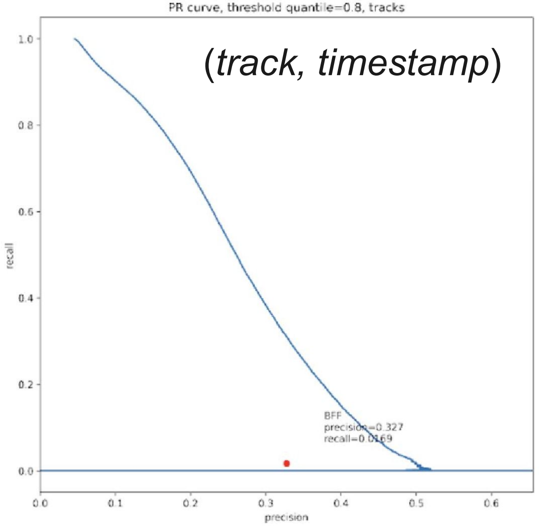


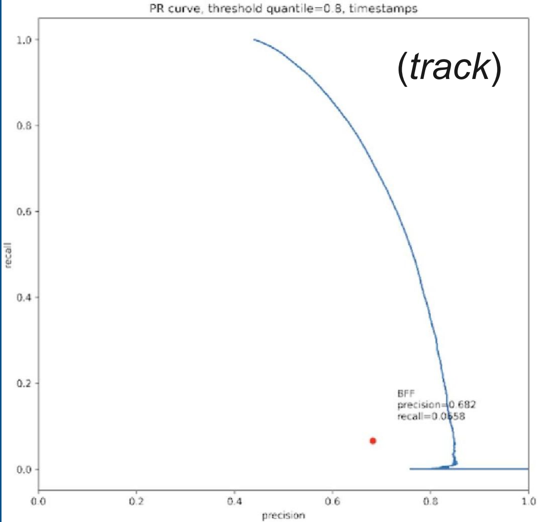


Result: TRE works **better** than MAF

Uncertainty: Precision vs Recall

- The closer to the ***upper right*** (higher precision and recall), the ***better***
- Results are kept for all the thresholds (***80-95%***)

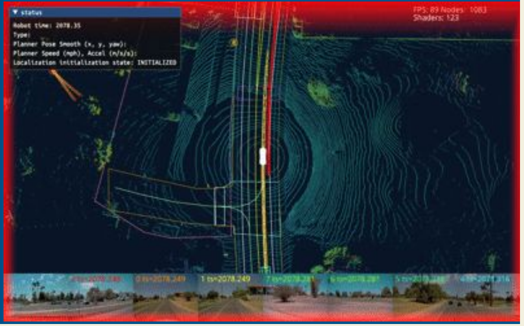




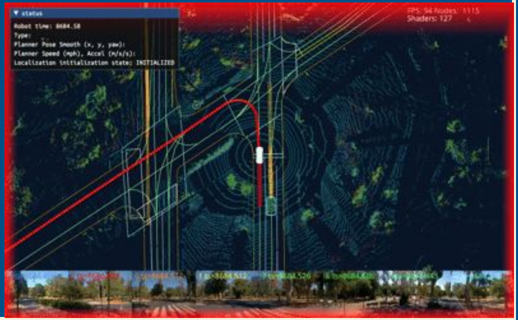
Result: TRE works **better** than MAF

Uncertainty Eval

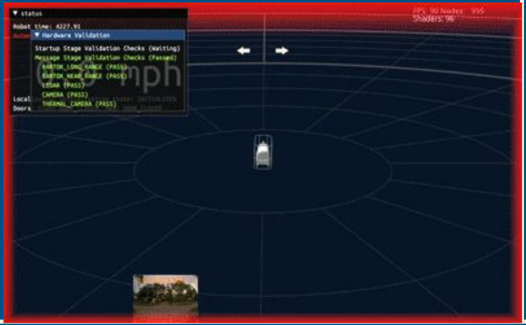
TP_1



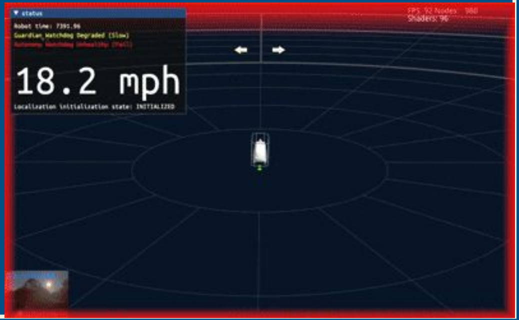
TP_2



FP

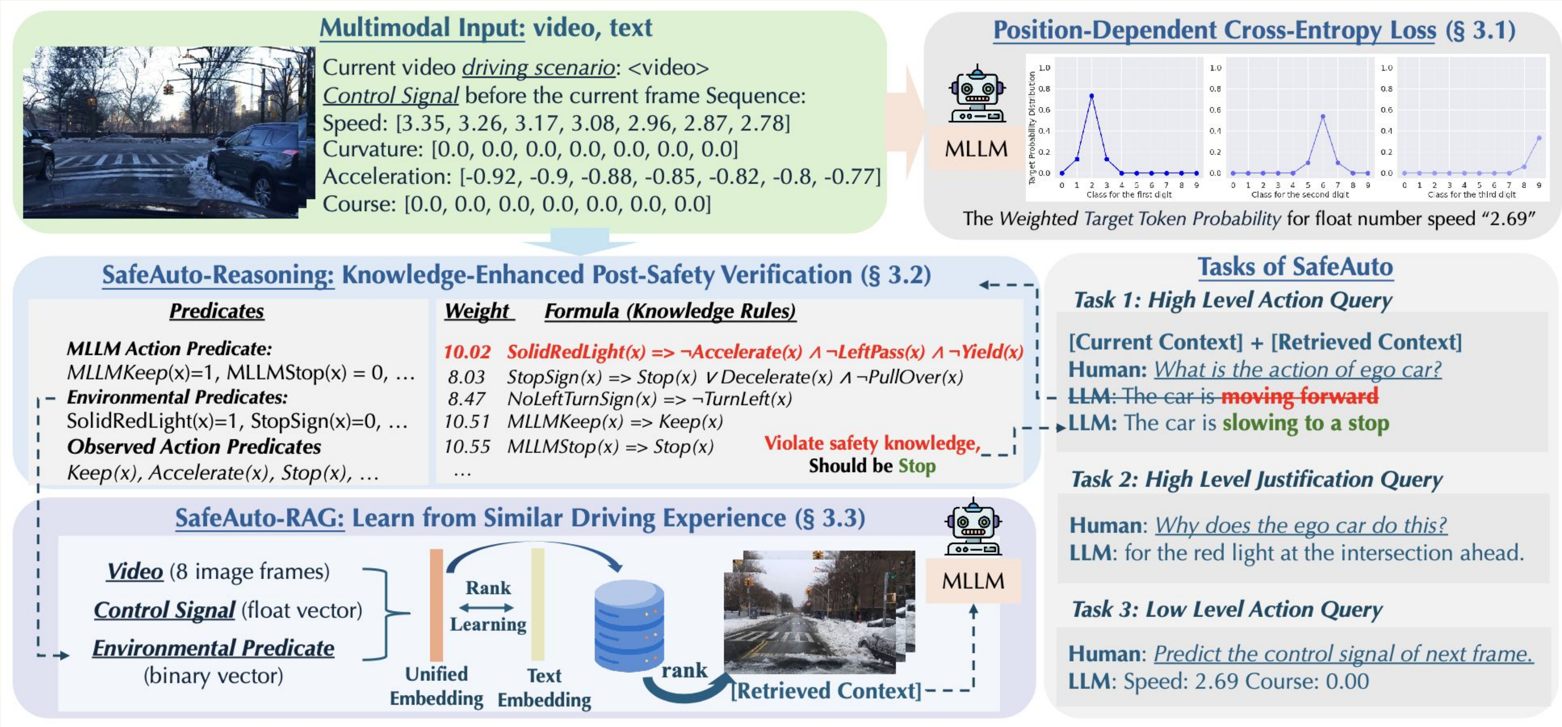


FN



1. Uncertainty and Safety
2. Uncertainty in AD: Prediction
3. **Safety in AD: Planning**
4. Conclusion

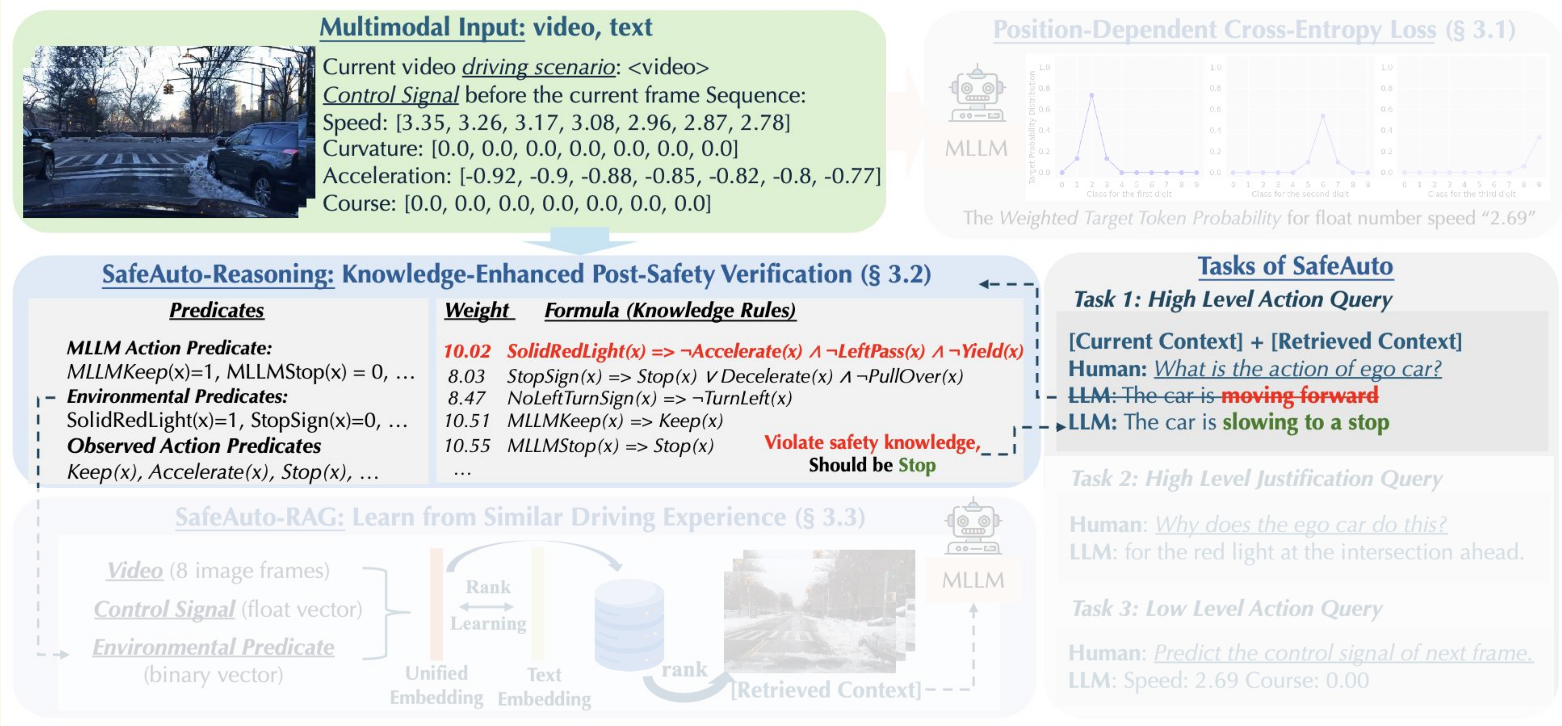
Safety: the Overall Approach [1]



***MLLM** = multimodal large language model*

[1] SafeAuto: Knowledge-Enhanced Safe Autonomous Driving with Multimodal Foundation Models

Safety: the Overall Approach [1]



***MLLM** = multimodal large language model*

[1] SafeAuto: Knowledge-Enhanced Safe Autonomous Driving with Multimodal Foundation Models

Safety: Motivations

- Currently, most MLLMs are still *data-driven*
- Reliability and strict adherence to safety regulations are inevitable
- Let's use *Probabilistic Graphical Models* to verify the safety
    - **Markov Logic Networks** (MLN) to combine:
        1. *Domain knowledge*
        2. *Traffic rules*

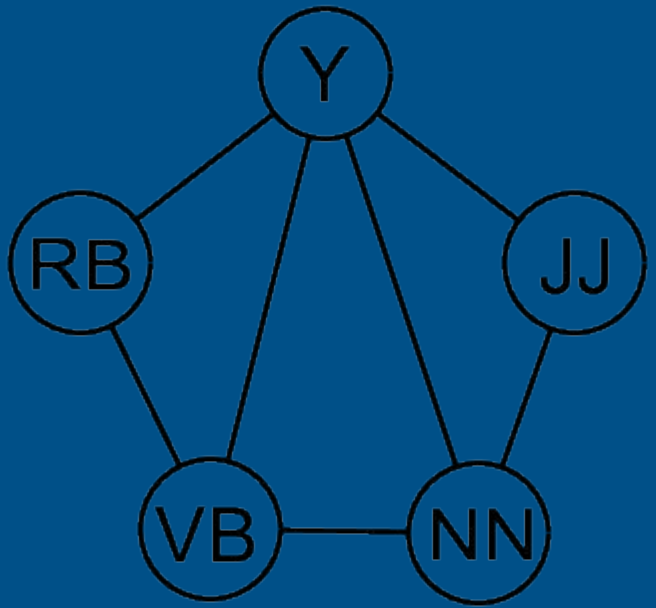


Image source

Safety: MLN

- MLN == a set of *first-order logic formulas* with an associated confidence *weight* ***w***
    - ***w***: to model uncertainty / deal with exceptions in real-world knowledge
    - Ex.: a traffic rule like "***If there is a stop sign, then the vehicle should stop or decelerate***" can be represented as the logical formula:
        - `StopSign(x)` ⇒ `Stop(x)` ∨ `Decelerate(x)`

$$P(\mathrm{X}) = \frac{1}{Z}\exp\left(\sum_{f\in F}\omega_f \sum_{a_f\in A_f}\phi_f(a_f)\right)$$

where:

- **X**: set of all ground truth predicates
- *Z*: partition function
- $\phi_f(a_f)$: potential function for formula *f* with assignment a_f (=1 iff a_f)
- *F*: set of all formulas *f*
- A_f: set of all possible assignments to the arguments of formula *f*

Safety: MLN in AD

- Predicates:
  - **Unobserved** *U*:
    - Vehicle should take (*Stop*, *Accelerate*, *TurnLeft*)
  - **Observed** *O*:
    - MLLM Action (*MLLMStop*, *MLLMAccelerate*, *MLLMTurnLeft*)
      - MLLMStop => Stop
    - Environmental (*StopSign*, *SolidRedLight*)
      - From video, using YOLOv8 [1] trained on LISA [2]
      - + Historical Control Signal (*HCSTurnLeft*)

$$\text{StopSign}(x) \Rightarrow$$

$$\Rightarrow \text{Stop}(x) \vee \text{Decelerate}(x) \wedge \neg\text{PullOver}(x)$$

Example of environmental *observed* predicate

[1] YOLO model
[2] LISA dataset

Safety: MLN in AD – Process (1)

- **Inference**
  - Obtain the most realistic *unobservable U* given the *observable O* using the trained *MLN*
- **Training**
  - Obtain the *weights* w_f to maximize the $P(U|O)$ with BDD-X [1] / DriveLM [2] data
- **Safety verification**
  - After inferring the *U* based on *O* from *MLN*, ***if*** it contradicts MLLM's action (a potential *safety violation* / *breach* of critical *traffic rules*) **=>** need to overwrite the high-level action query and *re-prompt* the MLLM *again*

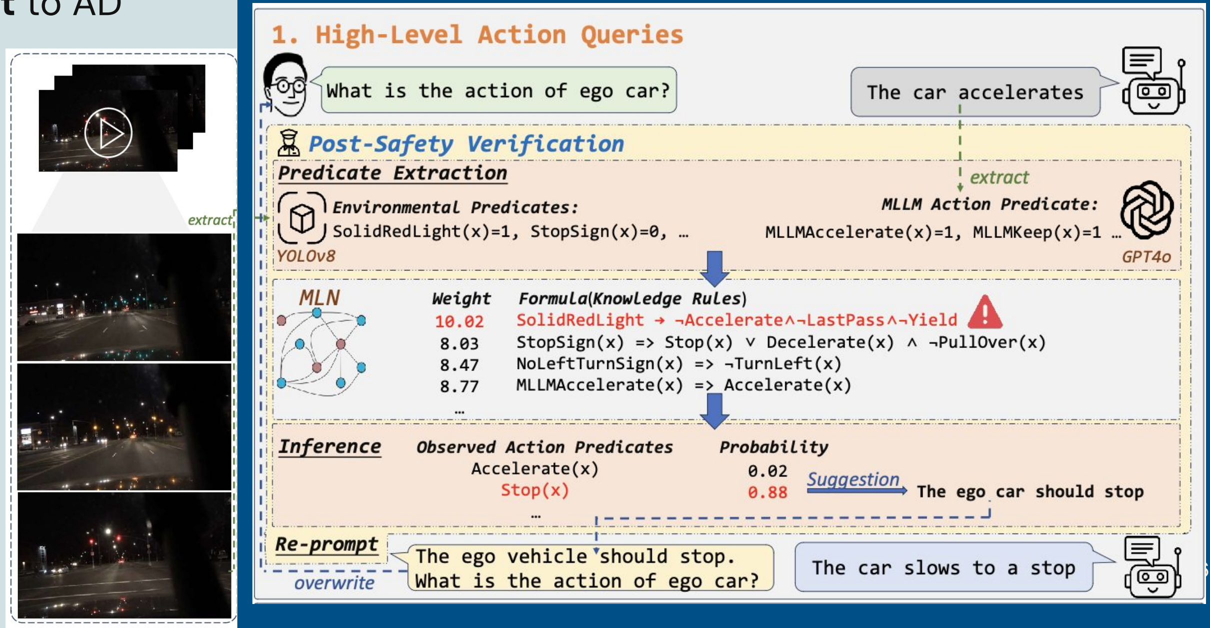
$$\mathcal{U}^* = \arg\max_{\mathcal{U}} P(\mathcal{U}|\mathcal{O})$$

$$\omega_f$$

[1] BDD-X dataset
[2] DriveLM dataset

Safety: MLN in AD – Process (2)

- **MLN**
  - Serves as a ***post-verification layer*** able to change the unsafe MLLM system initial suggestion
  - **Improving** the overall **trust** to AD system



Safety: MLN in AD – Results

- Ablation study on the **impact** of each module on the traffic rule violation rate of MLLM-predicted actions

Method	BDD-X	DriveLM
Base	11.64%	1.03%
PDCE	8.44%	1.46%
PDCE+RAG	5.90%	1.03%
PDCE+RAG+MLN	***4.50%***	***0.75%***

(lower the better)

DriveLM use case

Method	High-Level Behavior			Motion
	Accuracy	Speed	Steer	ADE
Base	60.58	64.57	80.29	0.86
PDCE	63.21	67.88	79.27	0.85
PDCE+MLN	***66.86***	***71.39***	***80.29***	***0.85***
PDCE+RAG	74.01	79.27	81.61	0.84
PDCE+RAG+MLN	***74.61***	***79.85***	***81.91***	***0.84***

1. Uncertainty and Safety
2. Uncertainty in AD: Prediction
3. Safety in AD: Planning
4. **Conclusion**

Uncertainty Outcomes

- EU / EBM model **works better** than the Bayes’ one
- Limitations:
  - Variability in the **ADE threshold**: objects, scenes, ts, etc
  - **No** any **time smoothing** for the OOD score
  - **One more** ML model
  - When the **ML model** becomes better, the training / eval **data** becomes **obsolete**

Safety Outcomes

- **Markov Logic Network** provides an additional layer of safety in AD
- Limitations:
  - Need to understand the **Markov**-based reasoning
  - Doesn't work **equally** best for every dataset
  - **One more** ML model

BDD-X use case

Method \ Metric	Action / Meteor	Action / Accuracy	Justification / Meteor
Base	29.2	61.75	13.2
PDCE	29.3	61.94	13.2
PDCE+MLN	***29.4***	***62.97***	***13.2***
PDCE+RAG	35.3	91.00	13.9
PDCE+RAG+MLN	***35.5***	***92.18***	***14.0***

Final Conclusion

- **Reliability**
  - Comes through the input-based **Epistemic Uncertainty**
- **Safety**
  - Achieved through the output-based correction by **Markov Logic Network**
- **Question** to be **Answered**
  - Can we combine everything at one and big (~foundation) model?

Thanks!