

# Uncertainty-Aware Autonomous Vehicles: Predicting the Road Ahead

Shireen Kudukkil Manchingal<sup>1†</sup>    Armand Amaritei<sup>2</sup>    Mihir Gohad<sup>2</sup>    Maryam Sultana<sup>1</sup>  
Julian F. P. Kooij<sup>3</sup>    Fabio Cuzzolin<sup>1</sup>    Andrew Bradley<sup>1,2</sup>

<sup>1</sup>School of Engineering, Computing and Mathematics, Oxford Brookes University, UK

<sup>2</sup>Autonomous Driving and Intelligent Transport, Oxford Brookes University, UK

<sup>3</sup>Cognitive Robotics, TU Delft, Netherlands

**Abstract**—Autonomous Vehicle (AV) perception systems have advanced rapidly in recent years, providing vehicles with the ability to accurately interpret their environment. Perception systems remain susceptible to errors caused by overly-confident predictions in the case of rare events or out-of-sample data. This study equips an autonomous vehicle with the ability to ‘know when it is uncertain’, using an uncertainty-aware image classifier as part of the AV software stack. Specifically, the study exploits the ability of Random-Set Neural Networks (RS-NNs) to explicitly quantify prediction uncertainty. Unlike traditional CNNs or Bayesian methods, RS-NNs predict belief functions over sets of classes, allowing the system to identify and signal uncertainty clearly in novel or ambiguous scenarios. The system is tested in a real-world autonomous racing vehicle software stack, with the RS-NN classifying the layout of the road ahead and providing the associated uncertainty of the prediction. Performance of the RS-NN under a range of road conditions is compared against traditional CNN and Bayesian neural networks, with the RS-NN achieving significantly higher accuracy and superior uncertainty calibration. This integration of RS-NNs into Robot Operating System (ROS)-based vehicle control pipeline demonstrates that predictive uncertainty can dynamically modulate vehicle speed, maintaining high-speed performance under confident predictions while proactively improving safety through speed reductions in uncertain scenarios. These results demonstrate the potential of uncertainty-aware neural networks - in particular RS-NNs - as a practical solution for safer and more robust autonomous driving.

## I. INTRODUCTION

Autonomous vehicle (AV) technology has made rapid strides, driven largely by advances in machine perception, *i.e.*, the ability of a vehicle to ‘see’ and interpret its environment. However, despite these gains, perception systems remain a critical bottleneck to AV safety. High-profile failures [1] have revealed that even state-of-the-art models can falter in ‘edge cases’: rare or out-of-distribution scenarios for which the system has no direct training experience [2]. A core issue is that most neural networks and modern perception systems are significantly over-confident in their predictions [3], [4], often providing a seemingly certain detection or classification even when there is substantial underlying uncertainty. This overconfidence is at odds with human driving behavior; a human will acknowledge that they are uncertain about the road ahead, and likely slow down, whereas a neural network typically offers a single ‘best guess’, with potentially life-threatening consequences.

<sup>†</sup> Corresponding author: [smanchingal@brookes.ac.uk](mailto:smanchingal@brookes.ac.uk)

\* This work has received funding from the European Union’s Horizon 2020 Research and Innovation program under Grant Agreement No. 964505 (E-pi).

The rise of autonomous racing competitions [5], [6], [7], emerged a critical testbed for advanced perception, planning, decision-making, and control algorithms to their limits; driven by stringent performance demands and safety constraints [8], [9], running on low-powered, lightweight hardware that must be inside the vehicle. High-speed autonomous vehicles (AVs) in racing scenarios must execute precise real-time predictions about their immediate environment while ensuring safety under rapidly changing conditions [10], [11].

Despite substantial research on robustness, both road-going and racing AVs remain susceptible to failures in the face of rare or unexpected events, such as novel weather patterns or unforeseen track layouts. These scenarios often go undetected as ‘risky’ by current systems, leading to overconfident actions at precisely the moments when caution is most warranted. In racing, this can result in an ‘off track’ event; on the road, it may have even graver consequences.

A key opportunity lies in equipping AVs with the ability to ‘know when they do not know’: to react with caution when uncertainty is high, or to push the performance envelope when it is appropriate to do so. Uncertainty-aware approaches to perception are an emerging area of research aimed at addressing exactly this need.

Traditional approaches to uncertainty quantification in computer vision typically rely on Bayesian neural networks [12], [13] or ensemble methods [14]. However, these approaches often struggle under severe domain shifts or when confronted with limited or ambiguous training data [15]. Recent advances have proposed explicitly modeling uncertainty via Random-Set Neural Networks (RS-NNs) [16], which predict belief functions over sets of possible outcomes, rather than a single probability distribution. RS-NNs circumvent the need to specify priors, as required by Bayesian methods, and do not require large training times, as in ensembles. They are also important in applications where the data is not too distinct as set relations can learn such nuances. Whilst promising approaches, to date such techniques have seen little application in the AV domain. In this work, we aim to fill this gap by evaluating uncertainty-aware models and integrating the most suitable approach into the AV software stack as shown in Fig. 1 (detailed in Sec. III-C), thus enabling real-time identification of ‘unexpected events’, in this case, a previously unforeseen road layout - allowing the vehicle to proactively reduce speed in the presence of uncertainty. To the best of our knowledge, this represents a novel application of uncertainty-aware modeling in autonomous driving,

directly embedded into the decision-making pipeline.

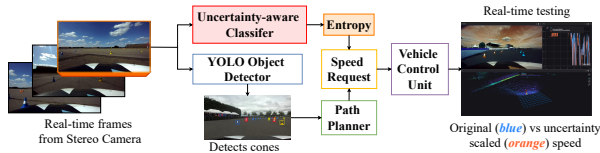


Fig. 1: Real-time uncertainty awareness in the autonomous racing car.

## II. RELATED WORK

Recent autonomous racing systems have pushed perception and planning to their limits, with teams like TUM and ETH Zurich demonstrating high-speed performance using CNN (Convolutional Neural Network) based cone detection, trajectory optimization, and minimum-curvature path planners [17], [10], [18], [19]. However, these pipelines remain highly sensitive to perception errors, motivating research into robustness through data augmentation and retraining [20], [21], [2], [11]. However, such methods lack explicit uncertainty quantification, leaving systems vulnerable in ambiguous or unseen scenarios.

Uncertainty quantification (UQ) has therefore emerged as a key factor in reliable AV perception, spanning segmentation [22], [23], [24], object detection [25], [26], [27], and depth estimation [28], [29]. Techniques such as Bayesian SegNet [22], MC dropout [12], evidential learning [30], and ensemble [14] methods have improved robustness under noise, occlusion, and OOD conditions. More recently, UQ has been extended to end-to-end controllers, enabling real-time steering angle uncertainty estimation [25]. Despite progress, major challenges hinder the full potential of UQ in autonomous vehicles. The field lacks ground-truth annotations, standardized evaluation criteria, and robust benchmarks, making it difficult to compare methods or assess real-world impact [31]. Many approaches also fail in open-world, out-of-distribution (OOD) scenarios. Most critically, UQ is rarely integrated into control or planning, serving mainly as a diagnostic metric.

Our work bridges this gap by posing next-turn prediction as a classification problem to enable quick, real-time results, and by evaluating uncertainty-aware models on this task. We integrate the best-performing model directly into an autonomous racing vehicle’s control pipeline, using its uncertainty estimates to dynamically modulate vehicle speed and enhance safety, robustness, and performance under realistic racing conditions. To the best of our knowledge, this is also the first classifier to predict the road layout ahead while providing calibrated uncertainty estimates, enabling proactive and uncertainty-aware planning.

## III. IDENTIFICATION OF ROAD LAYOUT UNDER UNCERTAINTY

### A. The Autonomous Racing Challenge

Autonomous racing competitions typically require a perception system to identify track boundaries, landmarks and other features. In this study, the vehicle (Fig. 2) competes in the Institution of Mechanical Engineers (IMechE) Formula

Student UK Artificial Intelligence competition<sup>1</sup>[7], where events are characterized by colored traffic cones that demarcate different track layouts. These cones must be identified and mapped, enabling a path to be planned so that the vehicle can complete a lap at the highest possible safe speed.



Fig. 2: The IMechE Autonomous Racing Car on track at Silverstone during Formula Student UK Artificial Intelligence.

### B. Identifying The Road Layout Ahead

For an autonomous race car, the ability to anticipate the upcoming road layout is fundamental for safe and optimal navigation. Unlike road vehicles operating at urban speeds, racing vehicles operate under conditions of high velocity, aggressive steering dynamics, and low tolerance for delay in decision-making. At racing speeds, small errors in perception can lead to large deviations in trajectory, resulting in loss of lap time or even safety-critical events.

To address this, we built a dedicated **TrackDrive Direction dataset** for this task (Sec. III-D). By identifying whether the road ahead is *straight*, *left-hard*, *left-medium*, *left-easy*, *right-hard*, *right-medium*, or *right-easy*, the vehicle gains the ability to:

- Plan steering inputs in advance for unexpected turns.
- Optimize speed and acceleration, e.g., slowing earlier for sharper turns (left-hard/right-hard) while maintaining higher speeds on straights or gentler curves.
- Reduce uncertainty in navigation, since classification of the upcoming track provides the control module with discrete, interpretable categories.
- Downstream reasoning with uncertainty measures.

**Classifying the Road Layout:** Our goal is to provide a high-level, discrete characterization of the upcoming track segment with uncertainty. Classification simplifies the interface between perception and control by mapping directly onto actionable decisions. For example, predicting ‘left-hard’ signals the planner to prepare for aggressive steering and braking, while ‘straight’ allows maximum throttle. Discrete predictions (e.g., ‘*Left-Medium*’ or sets like ‘{*Left-Medium*, *Left-Hard*}’) combined with uncertainty metrics (such as entropy) enable probabilistic reasoning and decision making. Although driving is continuous, we adopt discrete curvature classes because they map directly to stable control regimes and avoid noisy thresholding of small-angle regression errors. Moreover, epistemic ambiguity between adjacent maneuvers (e.g., {*Left-Medium*, *Left-Hard*}) is naturally expressed through set-valued classification.

**Uncertainty management** is important as road layout prediction is inherently affected by occlusions, variable

<sup>1</sup><https://www.imeche.org/events/formula-student/team-information/fs-ai>

lighting, out of sample road layouts etc. By incorporating categorical predictions with uncertainty estimates, the vehicle can reason probabilistically, for e.g., slowing slightly if entropy is high between two similar classes. Also *safety* is enhanced in autonomous racing, where dense cone layouts, narrow lanes, and changing conditions make accurate anticipation essential for avoiding surprises such as blind corners, thereby reinforcing robustness and overall performance.

### C. Using Uncertainty-aware Classifiers in the Autonomous Racing Car's Software Stack

The AV control system pipeline comprises perception, path planning, localisation & mapping and controls, implemented as individual nodes connected via ROS<sup>2</sup> [32]. These components work in tandem to find the right actuation command to send to the VCU given the layout of the track (Fig. 1). **Perception** is the entry point, as the car receives sensory information from the Zed 2i Stereo Camera and the Livox HAP TX LiDAR to identify cones in front of the vehicle. The data from the camera passes through the YOLOv11 Nano neural network [33] to identify the class of each cone (blue, yellow, small orange, large orange). Data from these two sensors is fused to create an accurate map in cartesian space of the track. The uncertainty-aware classifier runs alongside this process, predicting the curvature of the track from data received from the camera. Data is collected from testing runs with the Autonomous racing car. The training data was captured at an interval of 5 frames per second from previous recordings from the sensors of the AV. The positions of the cones are then passed to the localisation & mapping and path planning submodules, which plan a trajectory for the vehicle to follow given the landmarks. This sends an actuation command to the system controller (i.e. speed request and a desired steering angle).

Before the system controller passes the actuation command to the VCU, it tunes the received speed request using the uncertainty of the upcoming turn received from the classifier<sup>3</sup>, e.g., if a hard left turn is coming up but the classification of this turn from the classifier yields a high uncertainty, the speed request would be scaled down for the vehicle to excise caution until the classifier is sure that this is indeed a hard left turn, from which the correct speed request can be decided to navigate it.

### D. The TrackDrive Direction dataset

To achieve reliable classification of road layouts, a dataset consisting of seven classes was created, comprising a direction and severity of the upcoming track:

- **Straight**
- **Left Turn:** Left-easy, Left-medium, Left-hard
- **Right Turn:** Right-easy, Right-medium, Right-hard

This taxonomy reflects the natural variability in track curvature severity. Instead of attempting to regress continuous

curvature values (which are sensitive to small annotation errors), this discrete categorization makes the task more tractable and robust. In addition to the seven main classes, the dataset also includes three categories of **uncertain images** which are used for testing only: (1) **Random cone placement:** cones placed randomly without forming a proper track direction, (2) **Fallen cones:** cones lying on the track surface, creating ambiguity in the lane boundaries, and (3) **Confusing turns:** layouts where cones form intersections or ambiguous curves, making the intended driving direction unclear. These uncertain images are not used during training - instead they are solely for testing how well the model expresses uncertainty through measures such as entropy and mass functions assigned to sets in the presence of uncertainty. In total, the dataset contains 1,187 labelled images, of which 919 are standard class samples (split into training/validation/test with 20% of training data for validation) and 268 are uncertain cases reserved for testing model robustness. The dataset was collected by the racing team during multiple testing campaigns at two key locations: **Silverstone Circuit, UK:** <https://www.silverstone.co.uk>. **University Test Facilities:** Complementary data collected from controlled test tracks to balance samples and prevented model overfitting.

**Defining the Curvature Classes:** To categorize turns as 'easy', 'medium', or 'hard', we measure the track's deviation angle relative to the vehicle's forward direction (0°), represented by an orange arrow aligned with the car's nose (Fig. 3). The boundaries are:

- **Straight:** deviation angle < 15° (Fig. 3a).
- **Easy (Left/Right):** 15°–35° deviation (Fig. 3b). Requires only mild steering adjustments.
- **Medium (Left/Right):** 35°–60° deviation (Fig. 3c). Noticeable curve requiring sustained steering effort.
- **Hard (Left/Right):** > 60° deviation (Fig. 3d). Represents sharp bends or near-90° turns requiring aggressive steering and deceleration.

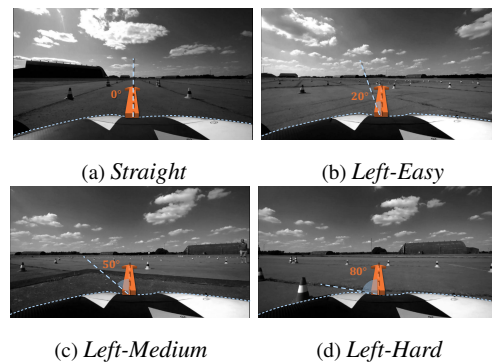


Fig. 3: Illustration of curvature classes: (a) *Straight*, (b) *Left-Easy*, (c) *Left-Medium*, (d) *Left-Hard*; right turns follow the same scheme. The orange arrow at the car's nose shows the forward axis, and the blue dashed line shows track deviation. Classes are defined by deviation: *Straight* < 15°, *Easy* 15°–35°, *Medium* 35°–60°, *Hard* > 60°. Images are black-and-white to highlight geometry.

## IV. EXPERIMENTAL RESULTS

We use a baseline *CNN*, a *RS-NN* [16], and a *Laplace Bridge Bayesian Neural Network (LB-BNN)* [13]. The Trackdrive dataset comprises 919 labeled images of steering directions,

<sup>2</sup><https://www.ros.org/>

<sup>3</sup>ROS-based code implementation will be released upon paper acceptance under an academic license.

with 735 for training (147 of these as validation) and 184 for testing. Additionally, 268 images labeled as ‘uncertain’ contain ambiguous or challenging scenarios to assess model robustness. All images are 224×224 pixels.

### A. Comparative Analysis of Model Performance

**Test accuracy:** We evaluated CNN, RS-NN, and LB-BNN on the TrackDrive dataset with seven classes: *Straight*, *Left-Easy*, *Left-Medium*, *Left-Hard*, *Right-Easy*, *Right-Medium*, and *Right-Hard*. Results (Tab. I) show clear performance differences: CNN achieved 67.93%, RS-NN outperformed with 75.54%, and LB-BNN lagged at 56.52%. RS-NN’s gain highlights the benefit of explicit uncertainty modeling, while LB-BNN’s limited accuracy suggests overly conservative predictions on small data.

Ambiguous classes such as *Right-Easy*, *Right-Medium*, and *Right-Hard* expose model weaknesses. CNN, relying on deterministic predictions, often misclassifies borderline samples. LB-BNN incorporates Bayesian parameter uncertainty but tends to under-generalize. RS-NN, in contrast, assigns belief masses across related sets (e.g., *Right-Easy* with *Right-Medium*, or *Left-Medium* with *Left-Hard*), reflecting structural similarities in data. This set-based treatment of uncertainty improves robustness, avoids overconfidence, excessive caution, and explains RS-NN’s superior accuracy.

We examine how models perceive each class during training by analyzing active learning behavior (§IV-E) under three acquisition strategies (§IV-E.1 to §IV-E.3) revealing overall accuracy growth (Figs. 10,13,16), per-class accuracy (Figs. 11,14,17), and per-class uncertainty (Figs. 12,15,18).

We tested a simplified 3-class version of the dataset: merging *Left-Easy*, *Left-Medium*, *Left-Hard* into a single *Left* class, and *Right-Easy*, *Right-Medium*, *Right-Hard* into a single *Right* class, leaving *Straight*, *Left*, and *Right*. Accuracies (Tab. I) show RS-NN again leading (87.50%), followed by LB-BNN (82.60%) and CNN (81.06%). The reduced performance gap reflects diminished class overlap after merging, and the gains for CNN and LB-BNN confirm that earlier struggles arose mainly from fine-grained distinctions in the seven-class task. All subsequent analyses are on the original 7-class scenario. While Deep Ensembles [14] are a strong uncertainty baseline, their multi-model memory and latency costs limit suitability for embedded racing hardware; our focus is single-model uncertainty for real-time deployment.

TABLE I: Test accuracy of CNN, RS-NN, and LB-BNN on the TrackDrive direction dataset. The **seven-class** scenario includes the classes: *straight*, *left-easy*, *left-medium*, *left-hard*, *right-easy*, *right-medium*, and *right-hard*.

Model	Accuracy (7-class)	Accuracy (3-class)
CNN	67.93 ± 0.25	81.06 ± 0.05
<b>RS-NN</b>	<b>75.54 ± 0.19</b>	<b>87.50 ± 0.19</b>
LB-BNN	56.52 ± 0.17	82.60 ± 0.23

### B. Uncertainty Estimation

Fig. 4 shows a comparative perspective of uncertainty estimation across the three different models. Each model’s predictions for regular test data are shown in solid colours and hatched bars represent predictions on uncertain test data.

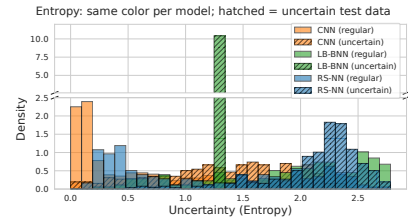


Fig. 4: Entropy distributions: *RS-NN*, *CNN* and *LB-BNN* on regular and uncertain (hatched) test data of Trackdrive dataset.

Notably, LB-BNN shows a pronounced spike at an entropy value near 1.3, suggesting a tendency towards conservative uncertainty estimates with moderate uncertainty for regular test samples. CNN entropy values are lower on average, implying generally confident but possibly overconfident predictions, reflected by lower entropy scores overall. Importantly, RS-NN presents a broader distribution of entropy scores across both regular and uncertain data. RS-NN entropy for uncertain data distinctly shifts toward higher values, clearly separated from regular data distributions, underscoring its effectiveness in identifying genuinely ambiguous scenarios. This ability to distinctly separate regular from uncertain predictions makes RS-NN particularly valuable for practical deployment where uncertainty-aware decision-making is crucial. RS-NN is the best performing model for accuracy and uncertainty, hence, we study the nature of its predictions in Sec. IV-C and apply it on a real-time AV in Sec. IV-D.

### C. Confusion Analysis and Uncertainty Characterization in RS-NN Model Predictions

We further analyze RS-NN performance by combining confusion matrices with entropy-based uncertainty estimation to identify strengths, weaknesses, and patterns of ambiguity. Key questions include:

- Which classes are most frequently confused, and what does this reveal about model performance?
- Effectiveness of entropy capture prediction uncertainty?
- Is directional confusion linked to uncertainty?

Fig. 5a shows the confusion matrix in test sample counts. Diagonals indicate correct predictions, with *Straight* most reliably recognized. Off-diagonals highlight in-directional confusions (*Left-Easy* vs. *Left-Medium*, *Right-Easy* vs. *Right-Medium*), marked by green/yellow rectangles. Less frequent inter-directional confusions (*Left*→*Right*, *Right*→*Left*), shown in red/purple rectangles, reflect natural ambiguity in edge cases. Fig. 5b complements this with pignistic entropy for each true-predicted pair. Low entropy values align with confident, correct predictions (e.g., *Straight* with mean entropy 0.61). Higher entropy accompanies ambiguous cases, such as *Right-Hard*→*Left-Easy* (2.31) or *Right-Hard*→*Right-Medium* (2.45). This confirms entropy as an effective quantitative measure of uncertainty.

Together, matrices reveal that RS-NN attain high accuracy and low uncertainty in distinct cases (*Straight*), but struggles with intra-directional subtleties (e.g., *Easy* vs. *Medium* turns). Entropy reliably highlights these ambiguous regions, showing uncertainty is more pronounced in directions than across them.

These insights suggest that RS-NN’s main challenge lies in distinguishing fine-grained turn variations.

Fig. 6 shows incorrect predictions, confirming most errors occur in closely related directional classes (e.g., *Right-Medium*↔*Right-Easy*, *Left-Easy*↔*Left-Medium*), further illustrating subtle feature similarities driving misclassification.

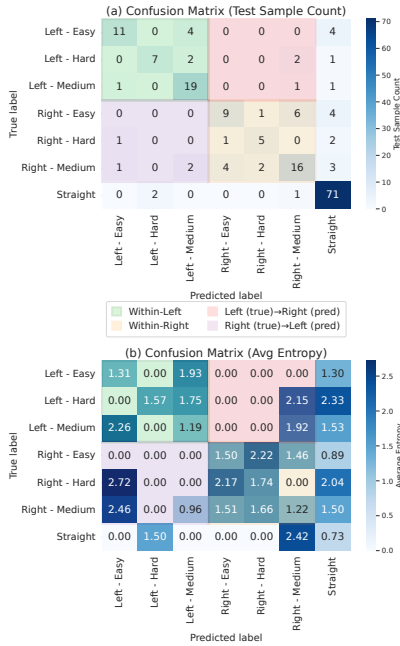


Fig. 5: Confusion matrices for RS-NN. (a) **Test sample counts**: frequency of correct and incorrect predictions across seven classes, with highlighted rectangles indicating within-group confusion (green, yellow) and between-group confusion (red, purple). (b) **Average entropy**: mean uncertainty for each true–predicted pair, where higher off-diagonal entropy reflects the inherent ambiguity and similarity between certain directional classes.

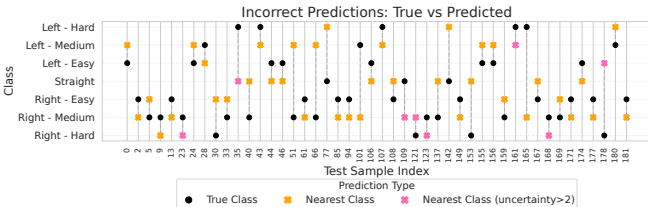


Fig. 6: Visualization of RS-NN misclassifications: black dots: true classes, crosses: nearest predictions (orange: low uncertainty, pink: high uncertainty). Dashed lines illustrate confusion between closely related classes.

Black dots represents the true class of a test sample, crosses indicate the nearest predicted class in the top mass set. The nearest class is determined by identifying the closest class included in the top mass set predicted by the RS-NN model. Dashed grey lines illustrate the proximity and confusion between closely related classes. The distinction between low uncertainty (orange crosses) and high uncertainty predictions (pink crosses) highlights the model’s capacity to recognise and quantify ambiguity effectively. High uncertainty cases (entropy > 2) underscore the difficulty of making precise classifications in particularly ambiguous scenarios.

#### D. Using RS-NN in Autonomous Racing Vehicle Control: Results

To validate RS-NN under realistic driving conditions, we integrated our uncertainty-aware classifier into the vehicle’s

software pipeline via ROS. The classifier runs inside of a dedicated ROS node which subscribes to the forward-facing camera, runs the RS-NN inference pipeline in real time, and publishes both the predicted Trackdrive direction class and its entropy on the predicted direction.

Fig. 7 showcases our real-time testing setup for the RS-NN model using Foxglove Studio (<https://www.foxglovestudio.co.uk/>). Left panel has live camera feed. Right panel displays a 3D LiDAR scan, with detected cones rendered as colored spheres.

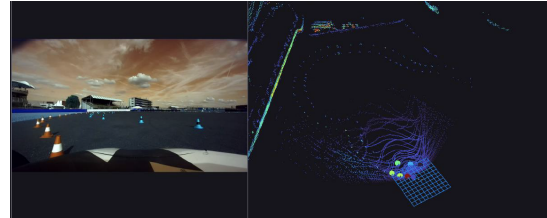


Fig. 7: Foxglove Test Dashboard: camera video (left), LIDAR scan (right).

The direction of the upcoming road layout is classified using the epistemic classifier - and the unique ability of RS-NN to ‘Know When it Does Not Know’ is then leveraged to intervene in uncertain scenarios - modulating the speed request in response to areas of high uncertainty in the track layout.

A five-tier speed-scaling policy was implemented, based on the RS-NN model’s predicted entropy at each timestep:

- Entropy < 2.2: run at full requested velocity (RPM<sup>4</sup>).
- 2.2 < Entropy < 2.3: reduce velocity to 90% (slight slow-down).
- 2.3 < Entropy < 2.4: reduce to 80% (moderate slowdown).
- 2.4 < Entropy < 2.6: reduce to 60% (stronger slowdown).
- Entropy ≥ 2.6: cut throttle entirely (full stop).

Thresholds were selected via empirical sweep testing to balance lap stability and conservatism; the non-uniform spacing reflects the observed nonlinear rise in control instability at high entropy. A learned entropy-to-throttle policy is left for future work.

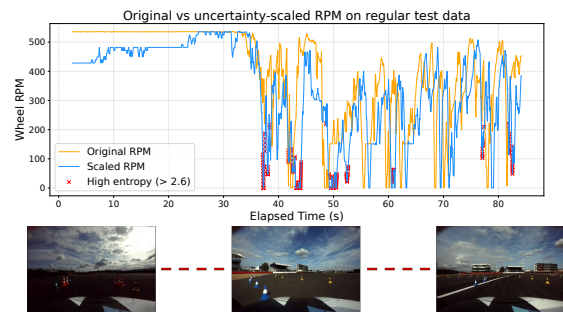


Fig. 8: Velocity traces for original vs. uncertainty-scaled RPM on regular track drive. Orange: planner’s requested RPM; blue: uncertainty-scaled RPM.

In Fig. 8, two throttle-related topics are overlaid in real time during a lap. The orange trace is the original RPM command from the high-level planner, while the blue trace is the adjusted RPM after applying our RS-NN uncertainty-based throttle controller. High-entropy moments where the RS-NN classifier reports entropy > 2.6 are highlighted with

<sup>4</sup>Wheel RPM: (revolutions per minute) is the rotational speed of a vehicle’s wheel, typically measured by wheel-speed sensors.

red ‘x’ markers. The plot shows that on straightaways and gentle curves (the first  $\approx 35$ s), entropy remains low, so the scaled RPM closely follows the original command. As the car approaches more complex turns ( $\approx 35$ -50s), isolated red crosses indicate brief spikes in uncertainty; here the blue line dips below the orange, enforcing slight slowdowns. Later in the run ( $\approx 50$ -80s), clusters of high-entropy points correspond to tighter or more ambiguous sections, triggering stronger RPM reductions and even momentary near-stops. Overall, the plot shows that on a familiar, regular course, uncertainty-driven scaling introduces targeted slowdowns only when and exactly where the model is least confident, otherwise preserving full throttle. This dynamic interplay clearly demonstrates how real-time uncertainty estimation can directly modulate vehicle behaviour, trading off speed for safety exactly when and where the perception model is least certain.

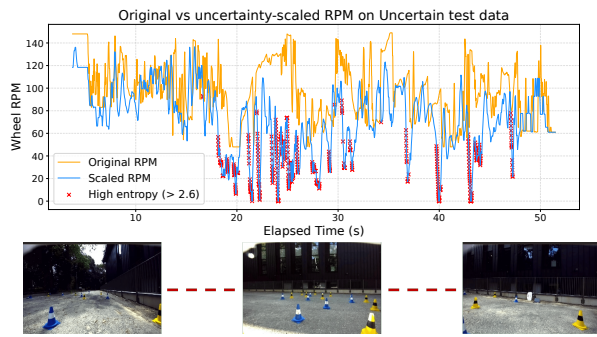


Fig. 9: Velocity traces for original vs. uncertainty-scaled RPM on uncertain (Random/Fallen Cone) track drive. Orange: planner’s requested RPM; Blue: uncertainty-scaled RPM.

Fig. 9 shows the same RPM comparison, but on a deliberately uncertain test track featuring random cone placements and fallen cones. Here, the classifier’s entropy frequently exceeds 2.6 (dense red ‘x’ clusters between  $\approx 15$ –45 s), signalling pervasive ambiguity in track geometry. Consequently, the uncertainty-scaled RPM (blue) is substantially and persistently lower than the original commands (orange), with frequent drops to near zero when entropy peaks. Unlike the smooth regular course, the scaled throttle now enforces extended slowdowns throughout the run, reflecting the system’s cautious response to unfamiliar or misleading cone layouts.

By overlaying the blue and orange signals, we can clearly see the controller’s safety envelope in action: proactive slowdowns or even stops in the face of uncertain perception, and full-speed operation when confidence recovers. This demonstrates how entropy-based control can translate model uncertainty into tangible, conservative driving behaviour; prioritizing safety whenever perceptual confidence is compromised.

### E. Active Learning

Active Learning (AL) improves model performance while reducing labelling costs by strategically selecting the most informative samples, often based on prediction uncertainty, rather than random sampling. This is particularly effective for uncertainty estimation, as AL targets ambiguous or borderline cases, improving decision boundary learning and model robustness, critical in safety-sensitive domains [34]. Our

experiments show that the RS-NN achieves superior results with less data, and we evaluate its performance through three active learning experiments using different initial sampling strategies and acquisition methods.

1) *Experiment 1: Stratified Initial Sampling with Uncertainty-based Acquisition:* We seed training with 10% stratified samples, reflecting natural class imbalance (dominated by Straight). This leads to initial overfitting on majority classes and poor minority performance. Subsequent acquisitions use entropy-based uncertainty, adding the most informative samples without correcting class imbalance. Results (Fig. 10): Both models start with low accuracy, but

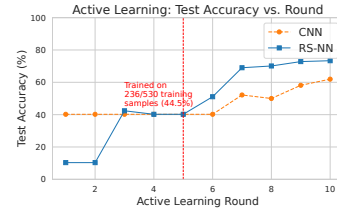


Fig. 10: *Experiment 1:* Test accuracy with 10% seeding: CNN improves slowly from 40%; RS-NN climbs faster and surpasses it by leveraging uncertainty.

RS-NN quickly outpaces CNN after 45% of data is labelled, highlighting the value of uncertainty-driven sampling. Class-level plots show Straight achieves high accuracy early, while minority classes improve later as more uncertain, class-specific samples are added; RS-NN consistently outperforms CNN. The group of plots in Fig. 11 presents the accuracy progression per class through the active learning rounds. Each subplot tracks one class: ‘Straight’ achieves near-perfect accuracy immediately, reflecting its dominance in the seed set and its distinct visual features. In contrast, minority turn classes (e.g., *Left-Hard*, *Right-Easy*) lag early on and improve only in later rounds as more uncertain, class-specific samples are acquired. In every case, RS-NN (blue) outperforms the CNN (orange).

Uncertainty trends (Fig. 12): RS-NN initially shows higher entropy, reflecting better-calibrated uncertainty, which decreases steadily as training progresses. CNN shows misplaced confidence early (low entropy despite poor accuracy), with entropy rising only later for harder classes. By the final round, RS-NN achieves both higher accuracy and lower uncertainty.

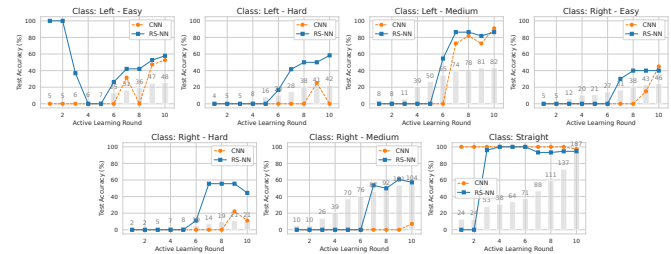


Fig. 11: *Experiment 1:* Per-class RS-NN vs CNN accuracy across active-learning rounds. *Straight* remains high from the start, while minority classes (e.g., *Left-Hard*, *Right-Easy*) lag early due to imbalance but improve as uncertain samples are added.

2) *Experiment 2: Balanced Initial Sampling with Class-wise Acquisition:* This experiment addresses the imbalance of Experiment IV-E.1 by starting with a balanced seed (five samples per class) and continuing with per-class uncertainty-based acquisition, ensuring equal representation across all classes.

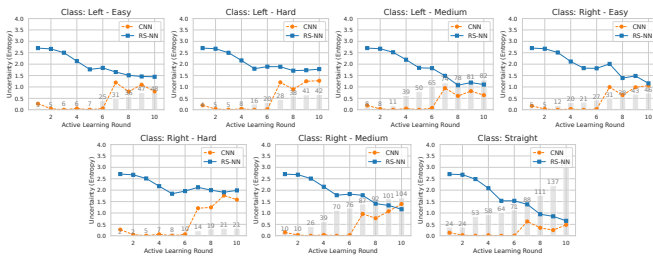


Fig. 12: *Experiment 1*: Per-class RS-NN vs CNN uncertainty (mean entropy) across active-learning rounds.

This strategy reduces bias, improves robustness, and enhances accuracy and uncertainty estimation. Under this balanced regime, RS-NN rapidly leverages uncertainty to outperform CNN (Fig. 13). Starting near 40%, RS-NN’s accuracy rises to  $\sim 70\%$  by 5 and plateaus near 73%, while CNN improves slowly and never catches up. This highlights RS-NN’s ability to efficiently exploit informative per-class samples. Class-level accuracy (Fig. 14) shows minority classes like *Left-Hard* and *Right-Hard* improving steadily from near-zero to above 50%, while majority classes (*Straight*, *Left-Medium*) achieve high accuracy quickly but still gain from targeted acquisitions. Grey bars confirm each class receives equal new labels per round. Uncertainty trends (Fig. 15) reveal RS-NN starts with higher entropy, particularly for ambiguous turns, reflecting better-calibrated caution. As per-class uncertain samples are added, entropy consistently declines alongside accuracy. CNN, however, shows misplaced early confidence (low entropy despite poor accuracy), only exhibiting higher entropy in later rounds. Overall, RS-NN demonstrates faster learning and superior calibration under active learning.

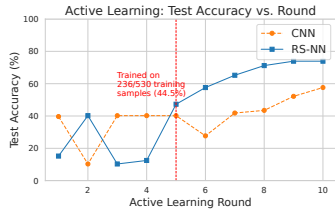


Fig. 13: *Experiment 2*: Test-set accuracy RS-NN vs CNN as a function of the number of labelled examples under a balanced seed and class-wise acquisition.



Fig. 14: *Experiment 2*: Per-class accuracy of RS-NN vs CNN under balanced acquisition.

3) *Experiment 3: Fixed Cumulative Class Targets*: This experiment enforces strict cumulative targets for labelled samples per class. Training begins with an equal number of examples (e.g., five per class) and increases uniformly each round (10, 15, then 20 per class), while acquisitions still prioritize uncertain instances. This fully balanced scheme demonstrates the upper bound of active learning performance, though it is less practical in real-world data-limited settings.

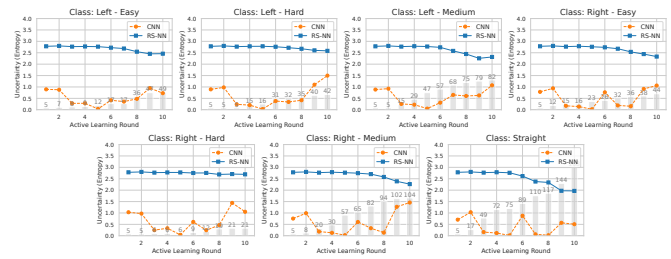


Fig. 15: *Experiment 2*: Per-class uncertainty (entropy) of RS-NN vs CNN under balanced acquisition.

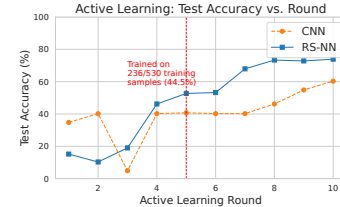


Fig. 16: *Experiment 3*: Test accuracy vs. active-learning round under fixed cumulative class targets (5 labels per class added each round). RS-NN (blue) consistently surpasses CNN (orange) by leveraging uncertainty sampling in a fully balanced schedule.



Fig. 17: *Experiment 3*: Per-class test accuracy under Experiment 3’s fixed per-class labelling schedule. Grey bars show labelled samples per class; accuracy for all classes improves steadily and predictably as more data are added equally.

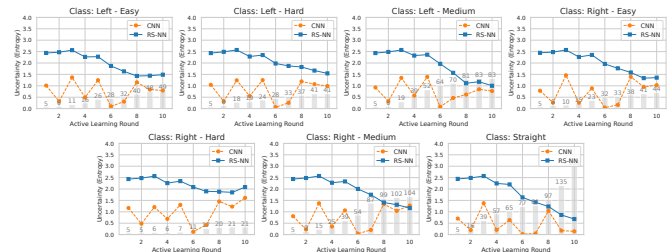


Fig. 18: *Experiment 3*: Per-class predictive uncertainty (entropy) across active-learning rounds. RS-NN (blue) begins with higher, honest uncertainty that steadily decreases, whereas CNN (orange) is overconfident early and only learns realistic uncertainty late in training.

Fig. 16 shows overall test accuracy under this protocol. With all classes represented from the outset, CNN starts near 40% once it has at least one sample per class, while RS-NN begins lower ( $\approx 15\%$ ) due to its cautious uncertainty estimates but quickly catches up. By round 4 (20 samples/class, total 140), RS-NN reaches  $\approx 75\%$ , surpassing CNN by 5–10 points and plateauing near 78%. Class-level results (Fig. 17) confirm steady improvements across all seven classes, including rare ones like *Right-Hard* and *Left-Hard*. *Straight* achieves near-perfect accuracy by round 3, while subtler turn classes (e.g., *Left-Easy*, *Left-Medium*, *Right-Medium*) exceed 80% only in later rounds. Grey bars verify that equal numbers of labels are added per class each round. Uncertainty analysis (Fig. 18)

highlights calibration differences: RS-NN begins with higher entropies ( $\approx 2.5$  nats) but reduces them consistently below 1.5 by round 6. CNN, by contrast, shows unrealistically low entropy ( $< 0.5$ ) early on, reflecting overconfidence, and only adjusts upward when harder examples appear. This again underscores RS-NN's advantage in both learning speed and calibrated uncertainty under active learning.

## V. CONCLUSIONS

This work evaluated uncertainty-aware models on an autonomous racing task. RS-NN consistently outperformed both a standard CNN and a Laplace-Bridge Bayesian network in accuracy and showed markedly superior uncertainty calibration. RS-NN's belief outputs and entropy estimates enabled more data-efficient active learning, rapidly improving minority-class performance under stratified and balanced sampling. Crucially, we validated RS-NN in a live AV control loop: by publishing real-time entropy via ROS, a multi-tier throttle-scaling policy directly translated predictive uncertainty into safer driving behaviour, by modulating speed according to entropy. This mechanism enforced caution in ambiguous scenarios (e.g., fallen cones, random layouts, poor visibility) while maintaining confidence-driven speed when conditions allowed. Thus, perceptual uncertainty was not only quantified but actively informed safer control decisions. Future work will extend this framework to uncertainty aware object detectors with multi-modal fusion (camera, LiDAR, radar). This work has demonstrated the potential of embedding uncertainty-aware perception into AV control pipelines, and aims to promote wider adoption of uncertainty-aware architectures - to usher in a new breed of safer, uncertainty-aware Autonomous Vehicles.

## REFERENCES

- [1] N. E. Boudette, "Tesla says autopilot makes its cars safer. crash victims say it kills." *International New York Times*, pp. NA–NA, 2021.
- [2] I. Fursa *et al.*, "Worsening perception: Real-time degradation of autonomous vehicle perception performance for simulation of adverse weather conditions," *arXiv preprint arXiv:2103.02760*, 2021.
- [3] Y. Ovidia *et al.*, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *Advances in neural information processing systems*, vol. 32, 2019.
- [4] M. Minderer *et al.*, "Revisiting the calibration of modern neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 682–15 694, 2021.
- [5] A. Nguyen, R. Nunn-Gage, A. Sepahmansour, and M. Sotoodeh, "Natcar," 2006.
- [6] V. Motevalli *et al.*, "Application of electric vehicle system design to grand prix ev kart," SAE Technical Paper, Tech. Rep., 2011.
- [7] D. E. Case, "Formula sae-competition history 1981-1996," SAE Technical Paper, Tech. Rep., 1996.
- [8] A. Wischniewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeyer, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle *et al.*, "Indy autonomous challenge-autonomous race cars at the handling limits," in *12th International Munich Chassis Symposium 2021: chassis. tech plus*. Springer, 2022, pp. 163–182.
- [9] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.
- [10] J. Kabzan *et al.*, "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.
- [11] H. Zhu, T. M. Tam Tran, A. Benjumea, and A. Bradley, "A scenario-based functional testing approach to improving dnn performance," in *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 2023, pp. 199–207.
- [12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059.
- [13] M. Hobbhahn, A. Kristiadi, and P. Hennig, "Fast predictive uncertainty for classification with Bayesian deep networks," in *Uncertainty in Artificial Intelligence*. PMLR, 2022, pp. 822–832.
- [14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [15] C. Guo *et al.*, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.
- [16] S. K. Manchinala *et al.*, "Random-set neural networks," in *The Thirtieth International Conference on Learning Representations*, 2025.
- [17] J. Betz *et al.*, "Tum autonomous motorsport: An autonomous racing software for the indy autonomous challenge," *Journal of Field Robotics*, vol. 40, no. 4, pp. 783–809, 2023.
- [18] S. Garlick and A. Bradley, "Real-time optimal trajectory planning for autonomous vehicles and lap time simulation using machine learning," *Vehicle System Dynamics*, vol. 60, no. 12, pp. 4269–4289, 2022.
- [19] L. Andresen *et al.*, "Accurate mapping and planning for autonomous racing," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 4743–4749.
- [20] I. Teeti, V. Musat, S. Khan, A. Rast, F. Cuzzolin, and A. Bradley, "Vision in adverse weather: Augmentation using cyclegans with various object detectors for robust perception in autonomous racing," *arXiv preprint arXiv:2201.03246*, 2022.
- [21] V. Muşat, I. Fursa, P. Newman, F. Cuzzolin, and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2915.
- [22] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.
- [23] Y. Zhao, W. Tian, and H. Cheng, "Pyramid bayesian method for model uncertainty evaluation of semantic segmentation in autonomous driving," *Automotive Innovation*, vol. 5, no. 1, pp. 70–78, 2022.
- [24] K. Sirohi, S. Marvi, D. Büscher, and W. Burgard, "Uncertainty-aware lidar panoptic segmentation," *arXiv preprint arXiv:2210.04472*, 2022.
- [25] S. Choi, K. Lee, S. Lim, and S. Oh, "Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6915–6922.
- [26] K. Okuz, T. Joy, and P. K. Dokania, "Towards building self-aware object detectors via reliable uncertainty quantification and calibration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9263–9274.
- [27] D. Feng *et al.*, "Labels are not perfect: Inferring spatial uncertainty in object detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9981–9994, 2021.
- [28] F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating scalable bayesian deep learning methods for robust computer vision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 318–319.
- [29] C. Qu, W. Liu, and C. J. Taylor, "Bayesian deep basis fitting for depth completion with uncertainty," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 147–16 157.
- [30] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, 2018, p. 3183–3193.
- [31] K. Wang, C. Shen, X. Li, and J. Lu, "Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications," *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [32] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science robotics*, vol. 7, no. 66, p. eabm6074, 2022.
- [33] A. Wong *et al.*, "Yolo nano: A highly compact you only look once convolutional neural network for object detection," in *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMCC2-NIPS)*. IEEE, 2019, pp. 22–25.
- [34] J. Knaup *et al.*, "Active learning with dual model predictive path-integral control for interaction-aware autonomous highway on-ramp merging," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 191–14 197.