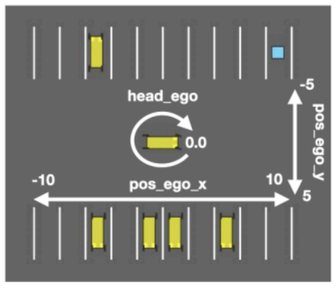
# Surrogate Reloaded: Fast Testing for Deep Reinforcement Learning with Bayesian Neural Networks

Rodrigo Montero González[1] supervised by Dr. Annibale Panichella[1], Antony Bartlett[1]

[1]EEMCS, Delft University of Technology, The Netherlands

## Background

Deep Reinforcement Learning (DRL) has enabled breakthroughs in fields like robotics, autonomous driving, and strategic games by allowing agents to learn complex behaviours through trial-and-error in simulated environments [1]. While DRL agents excel during training, testing their robustness remains a bottleneck. First, full simulation runs are computationally expensive. Secondly, failures are rare but critical, making them difficult to uncover through random testing. To address this, surrogate models have been proposed: instead of running full simulations, these models predict whether a given environment configuration will lead to failure, reducing testing cost. Previous work by Biagiola et al. [1] explored this idea using a Multi-Layer Perceptron (MLP) as the surrogate model. Building on their approach, we design a Bayesian Neural Network (BNN) surrogate model to predict failure cases in DRL environments and use it to guide a Genetic Algorithm (GA) in prioritising high-risk configurations without running full simulations. Our case study is based on the *HighwayEnv* simulator's Parking environment [2], where an autonomous vehicle must navigate into a goal lane while avoiding parked cars.




A

```
{
  "env_configuration": {
    "goal_lane": 20,
    "head_ego": 0.0,
    "pvehicles": {
       3, 5, 6, 8, 13
     },
    "pos_ego": (0.0, 0.0)
  }
}
```

B

*Fig:* Example configuration in the HighwayEnv Parking environment [2]. Adapted from Biagiola et al. [1].

## Research Questions

The following main research question guides the study:

***RQ1:*** *How do Bayesian Neural Networks compare to Multi-Layer Perceptrons as surrogate models for failure prediction in Deep Reinforcement Learning?*

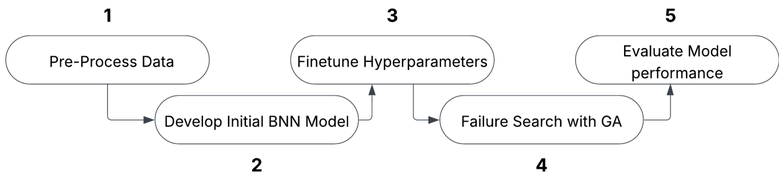
This study is further decomposed into two sub-questions:

***RQ1.1:*** *How do their predictive performances compare, particularly under class imbalance?*

***RQ1.2:*** *How effectively can BNNs guide Genetic Algorithm-based test generation toward discovering new failure scenarios?*

## Design methodology

Surrogate Modelling Workflow




1. **Data Preprocessing:** Reused data from Biagiola et al. [1], filtering for later DRL training stages to better simulate realistic agent performance. Applied an 80/20 train-test split.
2. **BNN Model Development:** Built Bayesian Neural Networks using the Blitz library, which supports scalable variational inference. Models use Gaussian priors and variational layers trained by minimising the Evidence Lower Bound (ELBO), composed of negative log-likelihood and KL divergence.
3. **Hyperparameter Tuning:** Conducted a grid search over 144 BNN configurations, spanning hidden layers, layer sizes, oversampling ratios, and imbalance strategies (class weighting, augmentation). Each configuration was trained across five seeds and ranked by validation AUC-ROC.
4. **Failure Search via GA:** Integrated the top BNN models into a Genetic Algorithm to guide test generation, replacing full DRL rollouts and enabling search over high-risk configurations more efficiently.
5. **Evaluation and Comparison:** Selected the best BNN based on GA failure discovery. Compared it to the MLP baseline using test set metrics, Mann-Whitney U test and Vargha-Delaney effect size. Also evaluated the diversity of discovered failures using entropy and coverage metrics.

## Conclusion and Future Work

Our findings show that BNNs are a promising alternative to traditional MLPs for failure prediction in DRL. BNNs achieved comparable generalisation in data classification and outperformed the MLP in discovering diverse failures when integrated into a Genetic Algorithm search. This highlights the value of uncertainty-aware surrogate models in enhancing DRL testing efficiency.

To strengthen generalisability and robustness, future work will extend the surrogate modelling framework to additional DRL environments such as the Lane Keeping and Humanoid environments.

## Results & Discussion

The classification metrics of the selected BNN and the MLP baseline are relatively similar, with the BNN achieving higher **accuracy**, **precision**, and **F1-score**, though the MLP has better **recall**. One-sided Mann-Whitney U tests confirm significant improvements for accuracy ($p = 0.004$) and precision ($p = 0.005$), with a borderline effect in F1 ($p = 0.069$). This suggests the BNN offers a more balanced trade-off between false positives and false negatives.

Table 1. Comparison on test set between MLP and selected BNN (1-layer, 64-hidden, augmented, class-weighted). Bold: best result.

| Metric | MLP | BNN |
|---|---|---|
| Accuracy | 0.784 ± 0.040 | **0.877** ± 0.014 |
| Precision | 0.110 ± 0.023 | **0.191** ± 0.034 |
| Recall | **0.477** ± 0.043 | 0.306 ± 0.071 |
| F1-Score | 0.178 ± 0.031 | **0.191** ± 0.034 |
| AUC-ROC | 0.697 ± 0.028 | **0.703** ± 0.011 |

In the GA-based failure discovery task, the BNN outperforms the MLP in both effectiveness and diversity. It discovers significantly more failures ($p = 1.4 \times 10^{-7}$), and its output diversity is substantially greater in terms of coverage ($p = 6.6 \times 10^{-15}$) and entropy ($p = 5.0 \times 10^{-6}$). Input diversity was comparable between the models, as both received the same configuration features, but the BNN ultimately uncovered a broader range of failure behaviours.

Table 2. Comparison of diversity and performance metrics between the MLP baseline and BNN across 50 GA runs. Input diversity is measured by coverage and entropy of input configurations; output diversity is measured from the failure outputs. Bold indicates a significant increase.

| Category | Metric | MLP | BNN |
|---|---|---|---|
| Performance | Failing Environments | 14.98 ± 3.24 | **19.14** ± 3.75 |
| Input Diversity | Coverage (%) | 50.00 | 52.00 |
| | Entropy | 0.00 | 1.17 |
| Output Diversity | Coverage (%) | 43.36 | **75.64** |
| | Entropy | 22.06 | **50.68** |

## References

[1] M. Biagiola and P. Tonella, "Testing of deep reinforcement learning agents with surrogate models," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 3, pp. 1–33, Mar. 2024, ISSN: 1557-7392. DOI: 10.1145/3631970.

[2] E. Leurent, "An environment for autonomous driving decision-making," *GitHub repository*, 2019, https://github.com/eleurent/highway-env.