Evaluating the Robustness of SAC under Distributional Shifts in Driving Domain

Lazar Polovina

Supervisors: Frans A. Oliehoek, Mustafa Celikok

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

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Lazar Polovina
Final project course: CSE3000 Research Project
Thesis committee: Frans A. Oliehoek, Mustafa Celikok, Annibale Panichella

Abstract

Reinforcement Learning (RL) has shown strong potential in complex decision-making domains, but its likelihood to distributional shifts between training and deployment environments remains a significant barrier to real-world reliability, particularly in safety-critical contexts such as autonomous driving. This study investigates the robustness of the Soft Actor-Critic (SAC) algorithm under such distributional shifts, with a focus on the influence of entropy regularization. Using the HighwayEnv simulator, SAC agents were trained with a range of fixed entropy coefficients as well as automatic entropy tuning. The agents were evaluated under varying traffic densities and environmental complexities. Experimental results reveal that moderate fixed entropy settings (0.05 and 0.2) each perform well under specific conditions, while a high entropy setting (0.9) achieves superior performance in more challenging scenarios. Notably, automatic entropy tuning consistently delivered the best overall results, achieving high average rewards and low crash rates across all test environments. All experiments were conducted on the DelftBlue supercomputer to ensure computational reliability and scalability. These findings underscore the importance of adaptive exploration strategies in improving policy generalization in the face of distributional shifts.

1 Introduction

Reinforcement Learning (RL) has emerged as a powerful framework for autonomous decision-making in complex and dynamic environments, such as autonomous driving and financial markets [12]. Despite achieving remarkable performance in many benchmark tasks, RL agents often suffer from reduced generalization when confronted with a distributional shift mismatch between training and testing environments [6]. This challenge is particularly concerning in safety critical applications like autonomous driving, where failure can have severe consequences [9].

A prominent class of RL algorithms used in continuous control is the Soft Actor-Critic (SAC), which enhances exploration by optimizing for both expected return and entropy [8]. SAC has shown strong empirical performance in environments where training and testing distributions are stable. However, recent work has questioned its robustness under distributional shifts [2; 5]. In particular, the entropy regularization coefficient, a hyperparameter central to SAC may play a crucial role in either enhancing or hindering generalization [11].

Given these concerns, this research aims to explore the following question:

> **How does the performance of SAC-trained agents degrade under increasing distributional shift, and how does this relate to the entropy regularization coefficient?**

To explore this question, we will assess the robustness of SAC in an autonomous driving simulation environment (`HighwayEnv`), using traffic density as a source of distributional shift. The investigation evaluates the influence of various fixed entropy coefficients on the robustness of the SAC algorithm. Furthermore, the effectiveness of automatic entropy tuning is examined as a mechanism for enhancing policy generalization under distributional shifts.

The remainder of this paper is structured as follows: Section 2 discusses related work on RL robustness and entropy regularization. Section 3 outlines our methodology, including the experimental setup and evaluation metrics. Section 4 presents and analyzes the results. Section 5 concludes with implications, limitations, and future directions.

2 Background

2.1 Reinforcement Learning

Reinforcement learning is a subfield of machine learning used for decision making in complex environments. The most widely used method for modeling reinforcement learning problems is by using the Markov Decision Process, a method in mathematics used for sequential decision making. The model that reinforcement learning uses is described by (S, A, P, R, γ) where S is a set of states, A set of actions, $P(s'|s, a)$ is the transition probability of reaching state s' from s if you take action a, R is the reward function and $\gamma \in [0, 1]$. There are various ways to solve reinforcement learning problems that are packed as Markov Decision Processes and one of them is to learn a value function usually denoted as $Q(s, a)$ [7]. This function estimates the expected cumulative reward that the agent will receive by taking action a from the state s. For small environments, these values can be stored in a table, known as a Q-table where each entry stores a reward value for a distinct action pair. On the other hand, if the environment has infinitely many state-action pairs, this approach becomes slow and very likely unusable, and that is why it is better to use approximate functions that can be modeled in various ways. Popular choice for function's model is deep artificial neural network, known as Q-network or Deep Q-Network. These networks take state as input and output Q-values or each action that the environment has.

2.2 Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC) is an actor critic algorithm that is used for maximum entropy reinforcement learning. Incorporating entropy into the actor-critic algorithm makes it more usable for environments with continuous action spaces, since entropy can be defined as:

$$\mathcal{H}(\pi(\cdot|s)) = -E_{a\sim\pi(\cdot|s)}\left[\log \pi(a|s)\right] \tag{1}$$

where $\pi(a|s)$ is the probability of selecting action a from state s under policy π. In SAC, stochastic policy $\pi_\theta(a \mid s)$ represents the actor that wants to maximize the expected return of the reward and entropy of the policies, and this can be seen from the equation:

$$J_\pi(\theta) = E_{s_t\sim\mathcal{D},\, a_t\sim\pi_\theta}\left[Q(s_t, a_t) - \alpha \log \pi_\theta(a_t \mid s_t)\right] \tag{2}$$

where Q(s, a) is the soft Q-function that estimates the expected return of taking action a from state s and α is the entropy regularization parameter, also known as temperature, that controls how much the agent will explore and exploit. While, the entropy coefficient α directly controls how much stochasticity is rewarded during policy optimization, the SAC policy remains inherently stochastic due to its parameterization. In practice, the policy $\pi_\theta(a \mid s)$ is modeled as a Gaussian distribution with a learned mean and variance, and actions are sampled from this distribution.

The critic component in SAC relies on two separately parameterized Q-value networks, both essential for the learning process. The use of two Q-functions helps reduce positive bias in value estimation and mitigates overestimation issues commonly seen in bootstrapped targets. The critic is trained by minimizing the Bellman residual:

$$J_Q(\phi_i) = E_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(Q_{\phi_i}(s_t,a_t) - y_t\right)^2\right], \quad i \in \{1,2\} \tag{3}$$

where the target y_t is computed as:

$$y_t = r_t + \gamma\, E_{a_{t+1}\sim\pi_\theta}\left[\min_{i=1,2} Q_{\phi'_i}(s_{t+1},a_{t+1}) - \alpha \log \pi_\theta(a_{t+1}|s_{t+1})\right] \tag{4}$$

In the equation above, γ is the discount factor and ϕ'_1 and ϕ'_2 are parameters of the target Q-networks.
This research will also involve automatic entropy tuning. The formula for automatically computing the entropy coefficient is as follows:

$$J(\alpha) = E_{a_t\sim\pi_\theta}\left[-\alpha\left(\log \pi_\theta(a_t|s_t) + \mathcal{H}\right)\right] \tag{5}$$

2.3 Distributional Shift

A main challenge when RL is deployed in the real world is when the test-time environment has some unexpected differences compared to training environment. These shifts are expected to occur in environments such as autonomous driving or the stock market when extreme conditions occur. Since RL agents are trained on fixed data sets, they have an exuse why they usually do not perform well when environment experiences something unusual and unexpected.

2.4 Motivation for This Research

This research questions whether changing the entropy coefficient during training can lead to improved robustness of the final policy.

3 Related Work

Soft Actor-Critic (SAC) has established itself as a foundational reinforcement learning algorithm for continuous control due to its maximum entropy formulation, which encourages exploration and stabilizes training [8]. Despite these strengths, its robustness under distributional shifts—common in real-world applications like autonomous driving or financial markets—remains a critical challenge.

Recent studies have explored SAC's vulnerabilities when deployed in unseen environments. Chen et al. [2] show that SAC can exhibit poor generalization in high-dimensional action spaces due to its disregard for test-time distribution shifts. Similarly, Enders et al. [5] proposed a risk-sensitive variant of SAC, emphasizing the need to incorporate robustness objectives into the policy optimization process.

Entropy regularization, a central component of SAC, has also received targeted attention. Ortal [11] investigated how different entropy settings influence robustness in autonomous driving tasks, finding that overly aggressive entropy tuning may lead to unstable behavior, while low entropy may hinder adaptability. Expanding on this, Massiani et al. [10] argued that entropy can be viewed as a viability-preserving mechanism, helping policies retain robustness by maintaining action diversity in uncertain scenarios.

To address robustness more formally, Cui et al. [3] introduced DR-SAC, a distributionally robust extension of SAC that explicitly accounts for model uncertainty. Their method adjusts the policy and value functions to be robust against worst-case distributions, outperforming standard SAC in high-risk environments.

The environment used in this study—Highway-env—was proposed by Bécsi et al. [1] and provides a suitable testbed for evaluating policy behavior under varying traffic conditions. Its configurability allows for realistic simulation of distributional shifts by altering lane counts, traffic densities, and driver models.

In financial applications, Sun et al. [12] proposed PrudexCompass to evaluate RL robustness under market dynamics, emphasizing the broader importance of assessing generalization outside of typical benchmark settings. Fujimoto et al. [6] further highlight that performance degradation due to distribution shift is not well captured by traditional RL metrics, calling for robustness-aware evaluation protocols.

Lastly, Homola [9] applied uncertainty-aware RL to flight control, demonstrating that similar robustness challenges exist in aerospace domains. These findings reinforce that the sensitivity of SAC to entropy and environment variability is not limited to driving, but rather a general limitation in current deep RL algorithms.

Taken together, these works underscore the importance of studying entropy coefficient tuning, environment variability, and robust policy objectives—precisely the focus of our investigation.

4 Methodology

For the investigation of the Soft Actor-Critic (SAC) robustness under distributional shifts, a set of controlled experiments within a simulated autonomous driving environment has been designed. The experimental environment is built on Gymnasium, a modernized framework for OpenAI Gym and for the implementation of Soft Actor-Critic model framework, stable-baselines3 is used because it is a reliable implementation of this algorithm and possesses an API that is easy to use.

4.1 Environment description

In order to evaluate the robustness of Soft Actor-Critic in autonomous driving environment, Highway-env is used. It is an open-source environment built with OpenAI Gym and it can simulate a multi-lane highway environment. It can also

offer both discrete and continuous sets of actions, and since we are estimating Soft Actor-Critic behavior we are going to choose continuous set of actions. To describe the simulation environment, an illustrative figure is included (Figure 1).

In this environment, the agent, represented as a green car, receives a reward based on multiple behavioral metrics.

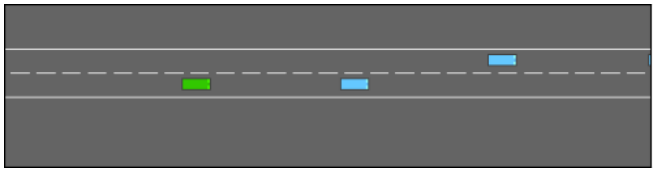

Figure 1: HighwayEnv

Specifically, the reward is a function of: the speed of the vehicle (R_y), the duration the agent remains in the correct lane (R_l), the time spent transitioning between lanes (R_v), and the agent's ability to maintain a safe (R_c) distance from surrounding vehicles [1]. These performance indicators are integrated into the reward function as follows:

$$R = \alpha_y R_y + \alpha_l R_l + \alpha_v R_v + \alpha_c R_c \tag{6}$$

where their corresponding weighting coefficients are equal to

$$\alpha_y + \alpha_l + \alpha_v + \alpha_c = 1 \tag{7}$$

4.2 Experimental setup

The experiment consists of a training and testing phase. In the training phase, the model is trained with a fixed entropy coefficient with a value within the range [0, 1], and it has a fixed traffic environment. During the training phase, the SAC agent is trained in a fixed traffic environment using a constant entropy coefficient, selected from the range [0,1]. Also auto entropy tuning is also used during training (see Appendix A.1). In the testing phase, different parameters are modified in order to answer the research question. The primary focus is on altering traffic density, but other environment parameters such as lane count and driver behavior modes are also varied to understand their influence on agent performance and decision making. That is why two environments for testing are going to be created, the one where all parameters except traffic density are going to be the same as the one used for training, and another one where parameters besides traffic density are going to be modified (see Appendix A.2) and the results of those two are going to be discussed in the next section.

All tests and training sessions for this research project were carried out on the Delft Blue supercomputer [4] and both phases used five different seeds (see Appendix A.3 and A.4).

4.3 Steps to perform the experiment

1. Train the model with Soft Actor-Critic algorithm.
2. Test Soft Actor Critic agent on this model by modifying various parameters from the highway-v0 model.
3. In the end, the total reward returned and the crash rate for various parameters are calculated and compared with other configurations.

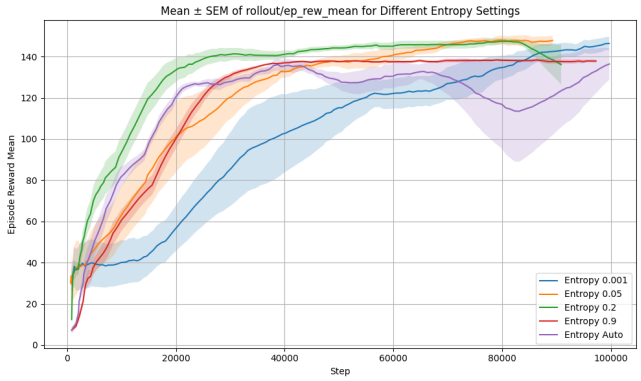

Figure 2: Training performance

4.4 Training

To evaluate the impact of entropy regularization on learning performance, SAC agents were trained under identical environmental conditions using five entropy configurations: fixed low (0.001), moderate (0.05 and 0.2), high (0.9), and adaptive (auto-tuned).

Figure 2 shows the progression of the mean episode reward during training. Among the tested configurations, the agent with entropy 0.2 achieved the highest overall performance and fastest convergence, suggesting that this moderate entropy setting provided an optimal balance between exploration and exploitation.

The auto-tuned entropy agent initially performed well and closely tracked the performance of the 0.2 agent, although it showed a late-stage dip, possibly due to over-adjustment or instability under changing policy dynamics. Entropy 0.05 also resulted in strong learning, albeit slightly below 0.2, and auto-settings.

The low entropy agent (0.001) demonstrated slow and steady improvement but consistently underperformed compared to other settings, indicating limited exploration. In contrast, the high-entropy agent (0.9) learned quickly early on but plateaued at a lower reward level, likely due to excessive randomness affecting convergence.

These results suggest that moderate fixed entropy values (especially 0.2) and adaptive tuning can lead to the most effective policy learning, while extremely low or high entropy values tend to either constrain exploration or introduce instability.

5 Discussion

This section presents a comprehensive analysis of the experimental results, focusing on how different entropy regularization strategies affect the robustness of Soft Actor-Critic (SAC) agents under distributional shifts.

In the first testing environment, where only traffic density was varied, the SAC agent trained with automatic entropy tuning achieved the highest average episode reward across all traffic levels, consistently outperforming all fixed entropy settings (Table 1, Figure 3). It not only achieved the highest rewards (e.g., 152.2 at 130 vehicles, see Appendix B.1), but also maintained very low crash rates, never exceeding 0.04 at

any traffic level. These results suggest that adaptive entropy tuning allows the agent to adjust its exploration dynamically, enabling it to generalize effectively under gradually shifting conditions without compromising safety.

Among the fixed configurations, entropy settings of 0.05 and 0.2 showed comparable performance, each outperforming the other at two out of four traffic levels. Specifically, 0.05 achieved slightly higher rewards at 10 and 130 vehicles, while 0.2 led at 30 and 70 vehicles. This indicates that both values are generally effective, but their optimality may vary depending on environmental complexity.

In contrast, the low-entropy agent (0.001) produced the lowest average rewards overall. However, its reward trajectory was non-monotonic—dropping significantly from 10 to 30 vehicles, recovering at 70, and falling again at 130—suggesting that extremely limited exploration can lead to unstable policy generalization under changing conditions.

Despite its poor performance, the 0.001 agent maintained a zero or near-zero crash rate across all traffic levels, with values ranging from 0.12 at 10 vehicles to 0.21 at 130. This implies that low entropy promotes overly cautious behavior that is safe but ineffective.

The high-entropy agent (0.9) consistently produced moderate rewards around 140 (see Appendix B.1) but was the only fixed-entropy agent to achieve a 0.00 crash rate across all traffic levels. This result defies the assumption that high entropy leads to erratic or unsafe policies; in this case, greater stochasticity likely promoted broader exploration and safer, more adaptive driving strategies.

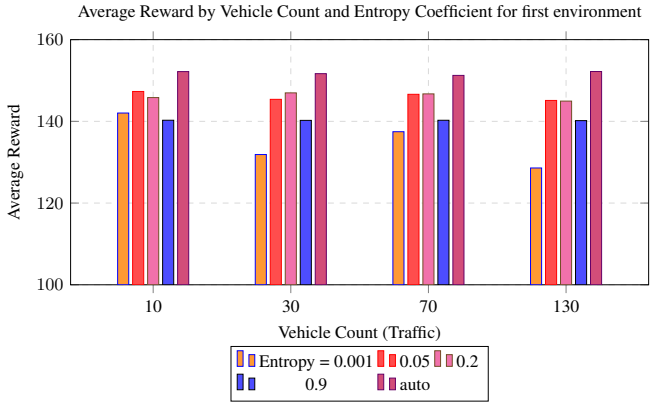

Figure 3: Comparison of SAC agent rewards across traffic levels with different entropy coefficients

Traffic	Entropy (Mean ± SE)				
	0.001	0.05	0.2	0.9	Auto
10	0.12 ± 0.07	0.03 ± 0.01	0.02 ± 0.02	0.00 ± 0.00	0.02 ± 0.01
30	0.22 ± 0.01	0.04 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01
70	0.17 ± 0.03	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01
130	0.21 ± 0.04	0.04 ± 0.01	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

Table 1: Crash rates across different traffic levels and entropy settings with standard errors for new environment

The second testing environment introduced additional complexity by reducing the number of lanes and modifying driver behavior to be more aggressive, creating a harsher distributional shift. Under these conditions, the auto-entropy agent again achieved the best performance (Figure 4) while maintaining very low crash rates across all traffic densities (Table 2), reinforcing its robustness in diverse scenarios.

Interestingly, the low-entropy agent (0.001), which previously had low crash rates, experienced a drastic spike in failure, with crash rates peaking at 0.71 at 70 vehicles. This sharp decline in safety under more complex conditions confirms that limited exploration can severely hinder adaptability and lead to brittle behavior.

The moderate entropy settings (0.05 and 0.2) again had a fine balance between reward and safety. However, in this more challenging environment, the high-entropy agent (0.9) performed better, it did not only have very low crash rates again, but also achieved higher rewards than 0.05 and 0.2 (see Figure 4 and Appendix B.2). This suggests that increased stochasticity becomes more beneficial as the unpredictability of the environment increases, enabling more resilient policy behavior.

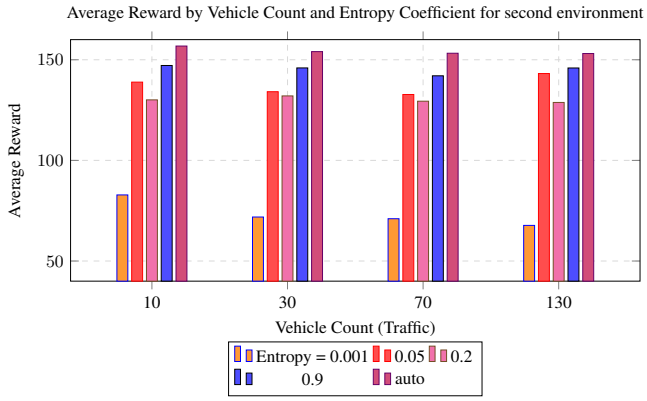

Figure 4: Comparison of SAC agent rewards across traffic levels with different entropy coefficients.

Traffic	Entropy (Mean ± SE)				
	0.001	0.05	0.2	0.9	Auto
10	0.64 ± 0.05	0.16 ± 0.03	0.16 ± 0.03	0.00 ± 0.00	0.03 ± 0.02
30	0.67 ± 0.06	0.11 ± 0.04	0.10 ± 0.05	0.01 ± 0.01	0.00 ± 0.00
70	0.71 ± 0.04	0.12 ± 0.03	0.12 ± 0.02	0.04 ± 0.03	0.02 ± 0.01
130	0.70 ± 0.06	0.04 ± 0.02	0.12 ± 0.02	0.01 ± 0.01	0.01 ± 0.01

Table 2: Crash rates across different traffic levels and entropy settings with standard errors for second environment

6 Responsible Research

This study was conducted entirely in a simulated environment (HighwayEnv) without involving human subjects or personal data, eliminating direct ethical risks. However, the broader application of RL in safety-critical domains like autonomous driving underscores the importance of robust policy evaluation under distributional shifts.

All experiments were run on the DelftBlue supercomputer using controlled random seeds and open-source tools (stable-baselines3 and HighwayEnv), ensuring reproducibility. Key

configurations such as entropy coefficients and traffic parameters are documented and can be shared upon request.

Limitations include the simplified nature of the simulation environment, which may not fully capture real-world driving variability. Efforts were made to minimize computational waste through efficient experiment design.

Throughout the development of this thesis, a large language model (ChatGPT) was used as a supportive tool for academic writing. Its assistance was limited to tasks such as improving clarity, refining structure, generating LaTeX code for appendices, and summarizing technical content. All scientific contributions, experiments, and interpretations presented in this work are the result of independent research. The use of the language model was guided by academic integrity, with the goal of enhancing communication quality, not replacing original thought.

7 Conclusions and Future Work

This research investigated how different entropy regularization strategies influence the robustness of Soft Actor-Critic (SAC) agents under distributional shifts in autonomous driving scenarios. While entropy coefficients play a key role in shaping exploration behavior, they are not the only mechanism governing the exploration-exploitation balance. In SAC, the stochastic nature of the Gaussian policy distribution inherently contributes to exploration, and this effect can be further influenced by action noise or other architectural factors.

Our experiments showed that while moderate entropy values (0.05 and 0.2) supported efficient learning in stable conditions, they did not generalize as well when agents encountered unfamiliar or more dynamic environments. Higher entropy (0.9) encouraged broader exploration and was better suited to more challenging, shifted environments, although this came at the cost of lower training performance. The most effective results across both stable and shifted test settings were achieved by the agent using automatic entropy tuning. This approach allowed the agent to dynamically adjust its exploration strategy based on policy uncertainty, which resulted in high average rewards and low crash rates.

The findings highlight that robustness in RL agents cannot be attributed to entropy settings alone, but rather to how these settings interact with the underlying policy structure and environment variability. Exploration in SAC is composed of both entropy regularization and stochastic policy.

Future research could explore more combinations of entropy control and noise injection strategies. In addition, incorporating larger training sets, using more agents, or using more realistic driving simulations would help validate these results. Investigating robustness under not constant or adversarial conditions and integrating strategies from risk-sensitive or distributionally robust reinforcement learning, may further improve the adaptability and reliability of SAC agents in real-world deployment contexts.

References

[1] Tamás Bécsi, Szilárd Aradi, Árpád Fehér, János Szalay, and Péter Gáspár. Highway environment model for reinforcement learning ∗∗the research reported in this paper was supported by the higher education excellence program of the ministry of human capacities in the frame of artificial intelligence research area of budapest university of technology and economics (bme fikpmi/fm).efop-3.6.3-vekop-16-2017-00001: Talent management in au-tonomous vehicle control technologies- the project is supported by the hungarian government and co-financed by the european social fund. *IFAC-PapersOnLine*, 51(22):429–434, 2018. 12th IFAC Symposium on Robot Control SYROCO 2018.

[2] Yanjun Chen, Xinming Zhang, Xianghui Wang, Zhiqiang Xu, Xiaoyu Shen, and Wei Zhang. Rethinking soft actor–critic in high-dimensional action spaces: The cost of ignoring distribution shift, 2025. arXiv:2410.16739v2.

[3] Mingxuan Cui, Duo Zhou, Yuxuan Han, Grani A. Hanasusanto, Qiong Wang, Huan Zhang, and Zhengyuan Zhou. Dr-sac: Distributionally robust soft actor-critic for reinforcement learning under uncertainty, 2025.

[4] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[5] Tobias Enders, James Harrison, and Maximilian Schiffer. Risk-sensitive soft actor-critic for robust deep reinforcement learning under distribution shifts, 2024.

[6] Ted Fujimoto, Joshua Suetterlein, Samrat Chatterjee, and Auroop Ganguly. Assessing the impact of distribution shift on reinforcement learning performance, 2024.

[7] Craig Gaskett, David Wettergreen, and Alex Zelinsky. Q-learning in continuous state and action spaces. In Ross Jeffery, editor, *Advanced Topics in Artificial Intelligence*, volume 1747 of *Lecture Notes in Computer Science*, pages 417–428. Springer, 1999.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

[9] Marek Homola. Uncertainty-aware reinforcement learning for flight control, January 2024.

[10] Pierre-François Massiani, Alexander von Rohr, Lukas Haverbeck, and Sebastian Trimpe. Viability of future actions: Robust reinforcement learning via entropy regularization. In *17th European Workshop on Reinforcement Learning (EWRL) 2024*. OpenReview, August 2024.

[11] Bartu M. Ortal. Evaluating robustness of deep reinforcement learning for autonomous driving: How does entropy maximization affect the training and robustness of final policies under various testing conditions?, June 2023.

[12] Shuo Sun, Molei Qin, Xinrun Wang, and Bo An. Prudex-compass: Towards systematic evaluation of reinforcement learning in financial markets, 2023.

A Configuration

A.1 Training Environment Configuration

```
config = {
    "lanes_count": 4,
    "vehicles_count": 20,
    "duration": 40,
    "other_vehicles_type": "highway_env.
        vehicle.behavior.IDMVehicle",
    "simulation_frequency": 15,
    "policy_frequency": 5,
    "observation": {"type": "Kinematics", "
        noise": 0.0},
    "action": {"type": "ContinuousAction"},
    "render_mode": None,
}
```

A.2 Evaluation Environment Configuration

Evaluation was performed under varying traffic levels defined by:

```
TRAFFIC = [10, 30, 70, 130]
```

Each configuration was evaluated independently with the corresponding ‘vehicles_count‘. The base configuration is as follows:

```
config = {
    "action": {"type": "ContinuousAction"},
    "lanes_count": 4,
    "vehicles_count": <value from TRAFFIC>,
    "duration": 40,
    "other_vehicles_type": "highway_env.
        vehicle.behavior.IDMVehicle",
    "simulation_frequency": 15,
    "policy_frequency": 5,
    "observation": {"type": "Kinematics", "
        noise": 0.0},
    "render_mode": None,
}
```

Second environment is as follow:

```
config = {
    "action": {"type": "ContinuousAction"},
    "lanes_count": 2,
    "vehicles_count": <value from TRAFFIC>,
    "duration": 40,
    "other_vehicles_type": "highway_env.
        vehicle.behavior.Aggressive",
    "simulation_frequency": 15,
    "policy_frequency": 5,
    "observation": {"type": "Kinematics", "
        noise": 0.0},
    "render_mode": None,
}
```

A.3 Soft Actor-Critic (SAC) Training Parameters

```
Policy: "MlpPolicy"
Entropy coefficient (ent_coef): [0.001, 0.05,
     0.2, 0.9, auto]
Total training timesteps: 100_000
Device: "cuda"
Training over 5 fixed random seeds: [0, 1, 2,
     3, 4]
Logging: TensorBoard, directory "./logs/
    sac_seed_<seed>"
```

A.4 Evaluation Procedure

```
- Ensemble of 5 SAC models trained with
    different seeds.
- Actions averaged across models at each
    timestep.
- Evaluation over 5 fixed random seeds: [42,
    43, 44, 45, 46]
- Metrics recorded:
    - Average reward (avg_reward)
    - Reward standard deviation (std_reward)
    - Success rate (1 - crash_rate)
    - Crash rate
- Results saved to: "results/
    eval_results_ensemble.csv"
```

B Results

B.1 Environment 1

Traffic	Average Reward – Environment 1				
	0.001	0.05	0.2	0.9	Auto
10	105.6	135.8	134.2	140.1	143.4
30	98.7	123.4	127.5	139.0	147.2
70	112.2	128.9	130.4	141.5	150.7
130	101.3	124.6	129.1	143.3	152.2

Table 3: Average rewards across different traffic levels and entropy settings for first environment.

B.2 Environment 2

Traffic	Average Reward – Environment 2				
	0.001	0.05	0.2	0.9	Auto
10	94.5	120.1	121.4	135.2	141.8
30	89.7	117.3	118.8	137.5	145.5
70	85.4	116.0	117.2	139.4	148.3
130	90.6	119.2	120.7	138.6	147.6

Table 4: Average rewards across different traffic levels and entropy settings for second environment.