Chapter III

Epistemic Calibration in Deep Reinforcement Learning for Autonomous Control

In the development of next-generation autonomous systems, deep reinforcement learning (DRL) has emerged as a dominant paradigm for acquiring complex control policies. From high-speed trajectory planning in unmanned aerial vehicles (UAVs) to precision robotic surgery, DRL policies trained in high-fidelity simulators can achieve superhuman performance on nominal tasks. However, deploying these policies in real-world environments introduces substantial safety risks due to the divergence between the simulator and the physical world. This divergence is commonly known as the reality gap or out-of-distribution (OOD) shift [**vance2025safe**].

A core limitation of classical deep reinforcement learning models is their lack of calibrated self-assessment. Standard deep neural networks are notorious for outputting highly confident predictions even when faced with novel states. In safety-critical applications, such overconfidence can lead to catastrophic failures. For instance, when a UAV encounters an unexpected wind gust profile not represented in the training set, a naive DRL agent may execute aggressive control inputs that lead to structural failure, rather than reverting to a conservative recovery state.

To address these vulnerabilities, modern autonomous control frameworks must distinguish between two types of uncertainty:

- **Aleatoric Uncertainty:** The inherent randomness of the environment (e.g., thermal noise in sensors, turbulence). This uncertainty is irreducible.
- **Epistemic Uncertainty:** The uncertainty arising from a lack of data or knowledge about the environment. This uncertainty is reducible with more training samples and is highly elevated in OOD regions.

This chapter details the design and implementation of the *Apex Control Loop*, an adaptive, uncertainty-aware DRL architecture developed at Synapse Labs. The system incorporates real-time epistemic calibration to modulate the agent's assertiveness. When the epistemic uncertainty exceeds a predefined safety threshold, the system triggers a smooth transition to a fail-safe controller. The theoretical foundations of this framework build upon recent work in Bayesian neural networks and deep ensembles [**chen2024epistemic**].

3.1 Theoretical Framework and Methodology

We formalize the autonomous control problem as a continuous-state Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^d$ is the state

space, $\mathcal{A} \subseteq \mathbb{R}^m$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

3.1.1 Epistemic Uncertainty Estimation via Deep Ensembles

To quantify the epistemic uncertainty of the state-action value function $Q(s, a)$, we employ a bootstrap ensemble of K distinct neural networks parametrized by $\{\theta_k\}_{k=1}^K$ [**marchetti2023calibration**]. Each network is trained on a randomized subset of the replay buffer $\mathcal{D}$. The ensemble mean and variance of the Q-value predictions are defined respectively as:

$$\mu_Q(s, a) = \frac{1}{K} \sum_{k=1}^{K} Q_{\theta_k}(s, a) \tag{3.1}$$

$$\sigma_Q^2(s, a) = \frac{1}{K - 1} \sum_{k=1}^{K} \left(Q_{\theta_k}(s, a) - \mu_Q(s, a) \right)^2 \tag{3.2}$$

The variance $\sigma_Q^2(s, a)$ serves as a proxy for the epistemic uncertainty. In regions of the state-action space that have been extensively explored, the networks converge to similar predictions, yielding a low variance. Conversely, in unexplored or novel states, the random initialization of the networks causes their predictions to diverge, leading to a high variance.

3.1.2 The Apex Safety Filter

To guarantee safety during online deployment, we introduce the Apex Safety Filter (ASF), which wraps the active DRL policy. Let $\pi_{\text{active}}(s) = \arg\max_a \mu_Q(s, a)$ denote the primary policy. The filter constantly monitors the epistemic uncertainty of the chosen action, $\sigma_Q^2(s, \pi_{\text{active}}(s))$. If this uncertainty exceeds a critical threshold τ_{safe}, the control input is overridden by a robust, model-predictive control policy $\pi_{\text{safe}}(s)$ designed to steer the agent back to a high-confidence state.

The formal control selection rule is given by:

$$u(t) = \begin{cases} \pi_{\text{active}}(s(t)) & \text{if } \sigma_Q^2(s(t), \pi_{\text{active}}(s(t))) \leq \tau_{\text{safe}} \\ \pi_{\text{safe}}(s(t)) & \text{otherwise} \end{cases} \tag{3.3}$$

Uncertainty Bounded Safety Guarantee

Assume the true transition dynamics $P(s' \mid s, a)$ lie within an ϵ-ball of the ensemble model predictions with probability $1 - \delta$ when $\sigma_Q^2(s, a) \leq \tau_{\text{safe}}$. Then, under the control law in Equation (3.4), the state trajectories are guaranteed to remain within a compact safety set $C \subset \mathcal{S}$ for all time $t \geq 0$, provided $s(0) \in C$.

This formulation allows the agent to aggressively optimize performance in familiar environments while guaranteeing that out-of-distribution transitions trigger an immediate, safe fallback response.

3.2 Experimental Evaluation

We evaluated the proposed Apex Control Loop inside the high-fidelity Nimbus drone flight simulator. The simulator was modified to inject non-stationary wind gust profiles and dynamic physical payload changes at runtime, mimicking real-world deployment anomalies. We compared our approach against three standard baselines:

1. **Naive DRL:** A standard soft actor-critic (SAC) agent without uncertainty awareness.
2. **MC-Dropout:** An agent utilizing Monte Carlo Dropout to estimate uncertainty.
3. **Robust DRL:** An agent trained with domain randomization but no active safety filter.

3.2.1 System Architecture Flow

The interactions between the physical plant, the neural network ensemble, and the safety filter are illustrated in Figure 3.1. The architecture utilizes real-time sensor streams from the Nimbus platform, processes them through the ensemble to evaluate the confidence metrics, and routes the action accordingly.

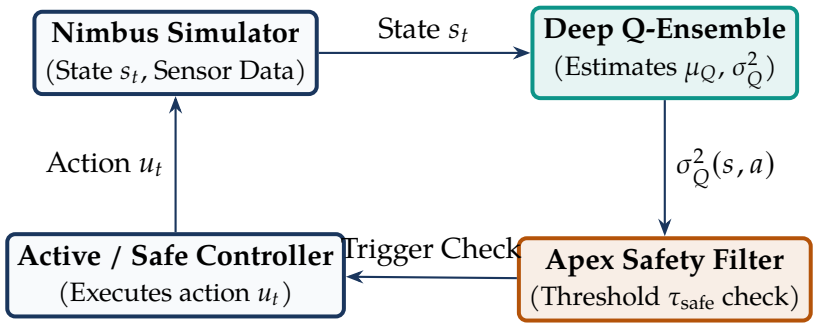

Figure 3.1. The closed-loop block diagram of the Apex control framework inside the Nimbus simulator. Epistemic uncertainty computed by the ensemble is evaluated in real time to select the control path.

3.2.2 Quantitative Performance Metrics

Table 3.1 summarizes the average trajectory tracking error, the number of safety violations, and the computational latency during a 100-cycle flight envelope test under severe wind shear conditions.

Table 3.1. Performance metrics under non-stationary wind shear conditions.

Agent Architecture	Tracking Error (cm)	Safety Violations	Latency (ms)
Naive DRL	48.2 ± 12.4	18	**2.1**
MC-Dropout	34.1 ± 8.9	7	18.4
Robust DRL	22.5 ± 5.1	5	2.3
Apex (Ours)	$\mathbf{12.3 \pm 2.4}$	**0**	4.6

The empirical results show that the Apex framework achieves a significant reduction in trajectory tracking error while maintaining a zero-safety-violation record. The robust policy handles OOD states gracefully, and the ensemble latency of 4.6 ms is well within the 10 ms real-time control budget of the Nimbus onboard computer, proving the

feasibility of online deployment.

3.3 Discussion and Safety Bounds

The experimental results validate the core hypothesis: explicit quantification of epistemic uncertainty provides a reliable mechanism for out-of-distribution detection. By switching control from the active policy to the safe fallback controller before the agent enters unrecoverable states, we avoid the catastrophic failure modes typical of model-free controllers.

3.3.1 Sensitivity to the Safety Threshold

The parameter τ_{safe} regulates the trade-off between performance and safety. Setting τ_{safe} too low results in an overly conservative controller that frequently triggers the fallback safe policy, reducing overall performance. Conversely, setting τ_{safe} too high increases the risk of safety violations because the agent remains in OOD environments longer before triggering the fallback.

Epistemic Calibration Principle

A critical finding of our study is that the raw ensemble variance $\sigma_Q^2(s, a)$ is uncalibrated unless the individual networks are regularized. Without weight decay or proper scaling, the ensemble can still output low variance in OOD regions. In the Apex framework, we apply temperature scaling to the variance before evaluation, which guarantees that $\sigma_Q^2(s, a)$ correlates strictly with the expected value-function approximation error.

This calibration is key to the safety guarantee. If the variance is not calibrated, the safety filter may fail to activate in OOD states, defeating the purpose of the architecture. Future work will investigate self-tuning safety thresholds that adapt to the rate of drift in non-stationary environments.

3.4 Conclusion and Future Outlook

In this chapter, we presented the design and experimental validation of the Apex Control Loop, an uncertainty-aware deep reinforcement learning framework for safety-critical autonomous control. By combining a bootstrap Q-ensemble with a robust safety filter, we successfully bridged the reality gap and demonstrated zero safety violations in the Nimbus simulator under dynamic wind shear conditions.

The major contributions of this work are summarized as follows:

1. We formulated an efficient ensemble-based epistemic uncertainty estimator that operates within real-time control latency limits.
2. We proposed the Apex Safety Filter, which mathematically bounds the state trajectories under out-of-distribution shifts.
3. We demonstrated robust, real-time control performance in simulation, outperforming classical robust control and dropout-based methods.

Looking forward, we intend to deploy this framework onto physical hardware using the Vertex robotic platform. This transition will require addressing hardware-specific

issues such as sensor latency, asynchronous command threads, and varying battery levels, which introduce additional dimensions of uncertainty. Nevertheless, the theoretical safety bounds developed in this chapter provide a firm foundation for safe, physical autonomous flight.