Uncertainty-Aware Reinforcement Learning Agents for Noisy Environments

Akash Singh
HEC, School of Management
University of Liège
Liège, Belgium
Akash.singh@uliege.be

Ashwin Ittoo
HEC, School of Management
University of Liège
Liège, Belgium
Ashwin.ittoo@uliege.be

Elise Vandomme
HEC, School of Management
University of Liège
Liège, Belgium
Elise.vandomme@uliege.be

Pierre Ars
Lead Actuarial Innovation
Ethias Insurance
Liège, Belgium
Pierre.ars@ethias.be

***Abstract*—Reinforcement Learning (RL) agents are highly sensitive to noise, particularly consecutive noisy states that destabilize training and can trigger catastrophic forgetting, a phenomenon inherent in real-world data as well. While uncertainty estimation has been widely explored for guiding exploration, its role in stabilizing value updates under noisy conditions remains relatively underexplored. In this work, we introduce MASURE (Masksembles for Stable and Uncertainty-aware Reinforcement Learning Environments), a novel framework that integrates Masksembles-based epistemic uncertainty into Q-learning. MASURE employs uncertainty-conscious value updates, leveraging the epistemic uncertainty to stabilize learning in noisy environments. We evaluate MASURE in both popular online RL benchmarks with sustained noise spanning consecutive states and in an offline real-world churn prediction task with inherently noisy features to test training stability. Across both settings, MASURE consistently improves stability and predictive performance, outperforming standard RL agents (DQN, BootstrapDQN) and state-of-the-art UE baselines (SunriseDQN, IVDQN). In noisy online benchmarks, MASURE achieves higher and more stable returns than IVDQN, while in the offline churn prediction task it attains the highest balanced accuracy (64.3%), surpassing DQN (63.5%), BootstrapDQN (63.8%), SunriseDQN (61.9%), and IVDQN (62.0%). Importantly, MASURE achieves these gains with significantly lower computational cost than deep ensembles, making it suitable for large-scale real world applications.**

***Index Terms*—Uncertainty, Epistemic, Masksembles, Deep Ensemble, Reinforcement Learning, Churn**

I. Introduction

Reinforcement Learning (RL) is a key AI paradigm that enables an agent to learn optimal behavior through trial and error, guided by rewards. Unlike supervised learning, RL is more suitable for environments where labeled data is scarce or unavailable. RL has gained much prominence in recent years, in robotics, self-driving cars, recommendations, finance, and resource management [1]. Most notably, RL (and Deep RL) are at the core of significant advances in AI, including chatGPT [2] (RL with Human Feedback) and AlphaGO [3].

However, RL methods can be sensitive to noisy environments, which destabilize learning which often leads to catastrophic forgetting, causing previously learned policies to be overwritten. [4]. These challenges limit the scaling of RL systems to real-world applications. To address these challenges, we propose a novel RL framework that stabilizes the learning process, enabling efficient training even in the presence of noise.

The core of our approach lies in estimating uncertainty of the states. In our formulation, uncertainty in a given state reflects the noise or perturbations in the observed state, e.g. inherent noise in real-world environments.

In our study, these uncertainty estimates are used to adjust the RL agent's value updates. Specifically, the higher the estimated uncertainty, the smaller are the corresponding updates. The underlying hypothesis is that noisy states lead an RL agent to misestimate the value of the state that has been perturbed with noise. This will lead to incorrect updates during learning (value or policy functions) such that the agent gets destabilized, forgetting all valuable states seen before (catastrophic forgetting).

Prior research has primarily exploited uncertainty to improve exploration [5]. In contrast, relatively little attention has been given to its role in stabilizing learning. Efforts to address stability have often relied on deep ensembles [6], [7]. However, deep ensemble approaches suffer from significant drawbacks. They require multiple forward passes and the training, evaluation, and inference of numerous neural networks. This becomes computationally very expensive as the ensemble size grows. Other RL methods with uncertainty, such as Bootstrapped DQN, also exist. However, they often suffer from correlation among ensemble heads, which reduces diversity and limits the reliability of their uncertainty estimates [7]. More efficient approaches like Monte Carlo (MC)-Dropout have been proposed, but they under perform when compared to deep ensembles [7], [8].

Our approach, described in section III, addresses these challenges by stabilizing the learning of deep RL(DRL) agents in noisy environments. Specifically, we adopt an architecture known as Masksembles [8]. Masksembles is an efficient uncertainty estimation (UE) neural network architecture that trains a single neural network using multiple parameterized binary masks [8]. By controlling the overlap between these masks, the method effectively creates an ensemble of diverse, uncorrelated sub-models within a single network, combining the benefits of deep ensembles and MC-Dropout. This method serves as a drop-in replacement, achieving performance on par with computationally expensive deep ensembles while

maintaining the efficiency of MC-Dropout [8]. Building on this foundation, we introduce **MASURE** (*Masksembles for Stable and Uncertainty-aware Reinforcement Learning Environments*), a novel architecture designed for uncertainty-conscious value update to stabilize learning under extreme noisy conditions. MASURE extends the Masksembles framework to RL, reducing computational overhead while improving robustness in both online and offline environments.

To evaluate the performance of our proposed method, we conducted two sets of experiments, ensuring generalizability:

- **Online setting:** We employed standard RL benchmarks, including *CartPole-v1*, *MountainCar-v0*, and *LunarLander-v2*. These environments are widely used in RL research because their simplicity and controlled dynamics make them well-suited for studying the impact of noise and the stability of learning. The choice of environments is inspired from recent state of the art work on uncertainty in RL [9] and builds on prior studies [7], [10], [11].
- **Offline setting:** We evaluated on proprietary real-world churn datasets from the insurance domain, where noise is inherently present in customer data.

We compared our approach against (i) popular baseline algorithms without explicit UE (DQN, BootstrapDQN [12]) and (ii) UE-based baselines (SunriseDQN, IVDQN [6], [9]). Performance was assessed using average episodic return in the online tasks and balanced accuracy (BA) in the churn prediction task. Across both settings, MASURE consistently outperformed the baselines under noisy conditions, achieving higher and more stable returns in online benchmarks and the best BA in offline churn prediction (64.3% vs. 63.5% for DQN, 63.8% for BootstrapDQN, 61.9% for SunriseDQN, and 62.0% for IVDQN).

II. Related work

Effective decision-making in real-world applications is often challenged by uncertainty and noise. The literature distinguishes between two main types of uncertainty in RL: aleatoric (inherent noise in the environment) and epistemic (uncertainty due to limited data/knowledge). The authors in [7] introduce a framework for disentangling aleatoric and epistemic uncertainty in RL, proposing four desiderata to capture the desired behavior for uncertainty estimates (UE) in RL setup. The desiderata cover aleatoric and epistemic uncertainty at both training and testing time for UE in RL models (e.g., MC-dropout, ensemble, evidential networks) inspired by supervised learning to instantiate these uncertainties. This distinction is crucial for improving exploration and stabilizing learning.

Several works focus on improving exploration efficiency by leveraging uncertainty estimates. For epistemic uncertainty and exploration, Liu et. al. [13] propose an approach to improve exploration efficiency for distributional RL inspired by Bayesian Deep Learning, using estimated epistemic uncertainty derived from ensembles of the quantile function network. Similarly, the study by authors of [6], is closely related to our work, proposes reweighting sample transitions based on uncertainty estimates from a Q-ensemble. They use an Upper-Confidence Bound (UCB) based on the mean and variance of Q-functions to select actions for efficient exploration. The work in [14] also points out gaps in ensemble methods and proposes an ensemble-diversified actor-critic algorithm for offline RL that leverages the generalization ability of deep neural networks and clipped Q-learning to penalize Out-of-Distribution (OOD) data points with high prediction uncertainties. The work in [9] introduces Inverse-Variance RL (IV-RL), a Bayesian framework combining probabilistic ensembles and Batch Inverse Variance weighting, which is currently a state-of-the-art method that we compare against.

Approaches for handling aleatoric uncertainty and noise are also prevalent. The authors of [15] addresses the issue where curiosity-driven exploration fails due to action-dependent noise sources ('noisy TVs') by proposing Aleatoric Mapping Agents (AMAs) that generate separate forward predictions for the mean and aleatoric uncertainty of future states to reduce intrinsic rewards for unpredictable transitions. The G-learning algorithm proposed by [16] tackles the slow learning problem in early stage Q-learning in noisy environments by regularizing the value estimates by penalizing deterministic policies.

Other approaches include [17], which proposes Noisy Nets as an extension for various Deep RL (DRL) algorithms (like DQN, A3C) to yield better performance through efficient exploration. Furthermore, the authors in [18] proposes filtering out noisy samples during training by employing the variance of the samples. More complex techniques include [19] use of Reward Machines for Deep RL in noisy and uncertain environments to leverage task structure and prior knowledge. The work by [20] introduces a novel DRL algorithm for uncertainty and noise-aware decision-making using Bayesian Neural Networks and skew-geometric Jensen-Shannon divergence. Investigating the statistical properties, the authors in [21] study the central limit theorem behaviors of estimated Q-values and value functions to construct asymptotically valid confidence regions. Finally, the authors in [22] proposes recursively learning the uncertainty of the Bellman equation to solve for an over-approximation that may lead to inefficient exploration.

III. Proposed Uncertainty aware RL method

Our approach builds upon recent advances in UE-based RL. In particular, we integrate uncertainty-based sample reweighting, exploration strategies, and reward shaping, drawing inspiration from prior work such as [6], [9], and [23].

A. Masksembles

For UE, a wide range of methods have been proposed in machine learning and RL like Monte Carlo (MC) dropout [7], [24], deep ensembles [7], [25], deep kernel learning [7], [26], and evidential networks [7], [27].

Deep ensembles estimate predictions by averaging outputs from independently trained models, where aleatoric uncertainty is captured by the average of predicted variances σ_i^2 and epistemic uncertainty from the variance of mean predictions

μ_i. Although highly effective, this approach is computationally expensive as training, evaluation, and inference scale with the number of ensemble members [7], [8]. In contrast, MC-Dropout provides a cheaper approximation by running a single trained network K times with different random dropout masks, combining the outputs to compute the mean prediction, aleatoric uncertainty from predicted variances σ_k^2, and epistemic uncertainty from the variance of mean predictions μ_k. Despite their computational efficiency, MC-Dropout often underperforms relative to deep ensembles, since simply adding randomness does not always yield diverse predictions [8].

To overcome the above mentioned gaps, Masksembles [8] were proposed as a new architecture combining the strengths of both approaches. Instead of random dropout, Masksembles employ a fixed, predefined set of binary masks with controlled overlap. This reduces correlation across sub-models, ensures diversity, and yields uncertainty estimates that match deep ensembles at significantly lower computational cost (Table I) [8]. We leverage these properties in MASURE, where the variance across Masksemble heads quantifies epistemic uncertainty to stabilize learning under noise. MASURE is instantiated on a Q-learning backbone (section III-B) and formalized through an uncertainty-conscious update rule (section III-C).

TABLE I
COMPUTE EFFICIENCY COMPARISON OF MASURE AND ENSEMBLE-BASED BASELINES (OVER LUNARLANDER-V2).

Model	Params	Inference FLOPs	Training FLOPs
DQN	5.00K	9.86K	24.64K
SunriseDQN (Deep Ensemble)	24.98K	49.28K	123.20K
IV-DQN (Deep Ensemble)	26.28K	51.84K	129.60K
MASURE (Ours)	**6.02K**	**36.69K**	**91.72K**

B. Q-Learning and Contextual Bandits

We denote the state by s (possibly vector-valued), the action by a (discrete), and write $Q(s, a)$ for their state–action value.

The standard Q-learning update is defined as:

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha \cdot \delta_t \tag{1}$$

where δ_t denoting the temporal difference(TD), defined as:

$$\delta_t = r_t + \gamma \cdot \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t) \tag{2}$$

and where α is the learning rate, r_t is the reward received at time t, $\gamma \in [0, 1)$ is the discount factor, and s_{t+1} denotes the next state. In a one-step contextual bandit setting, there are no transitions ($\gamma = 0$). The update therefore simplifies to:

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha \big(r_t - Q_t(s_t, a_t)\big). \tag{3}$$

We frame the customer churn prediction problem as a contextual bandit. At each time step, the agent observes a context vector x_t (customer profile and behavior) and chooses an action a_t from a discrete set (e.g., "Churn," "Not Churn"). The agent then receives an immediate reward r_t, representing customer retention or churn. The goal is to learn a policy $\pi(a|x)$ that maximizes total expected reward, thereby classifying churn.

While these Q-learning provide the learning backbone, they do not account for the destabilizing effect of noise. We therefore propose MASURE, which integrates Masksemble-based uncertainty estimation directly into the Q-learning update. The key idea is to modulate the update magnitude according to epistemic uncertainty, thereby dampening unstable updates in noisy states.

C. MASURE: Masksembles for Stable and Uncertainty-aware RL Environments

We formalize MASURE with an uncertainty-conscious update designed to stabilize Q-learning under consecutive noisy states by scaling the temporal difference step inversely with epistemic uncertainty. Real-world data is inherently noisy, and such noise induces epistemic uncertainty that can destabilize learning. MASURE employs DQN [28] as its backbone, estimating epistemic uncertainty from variance to adaptively scale Q-value updates. We use an ensemble of n Masksemble subnetworks, where each mask $i \in \{1, \ldots, n\}$ defines a sub-model that produces a Q-value estimate $Q_t^{(i)}(s_t, a_t)$. The ensemble mean estimate is:

$$\hat{Q}_t(s_t, a_t) = \frac{1}{n} \sum_{i=1}^{n} Q_t^{(i)}(s_t, a_t). \tag{4}$$

and the epistemic uncertainty is given by the empirical variance across subnetworks:

$$\sigma_t^2(s_t, a_t) = \frac{1}{n} \sum_{i=1}^{n} \big(Q_t^{(i)}(s_t, a_t) - \hat{Q}_t(s_t, a_t)\big)^2. \tag{5}$$

The Bellman target is defined as:

$$y_t = r_t + \gamma \cdot \max_{a'} \hat{Q}_t(s_{t+1}, a'). \tag{6}$$

with temporal difference error:

$$\delta_t = y_t - \hat{Q}_t(s_t, a_t). \tag{7}$$

Finally, our proposed uncertainty-conscious Q-value update is:

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow \hat{Q}_t(s_t, a_t) + \frac{1}{1 + \sigma_t^2(s_t, a_t)} \big(\alpha \cdot \delta_t\big) \tag{8}$$

where α is the learning rate.

The update is made uncertainty-conscious by scaling it inversely with the epistemic variance $\sigma^2(s, a)$, so that higher uncertainty results in smaller effective updates to $\hat{Q}(s, a)$. This mechanism prevents large, unstable changes during consecutive noisy states and improves stability in both contextual bandits and full RL environments. The $1/(1 + \sigma^2(s, a))$ is chosen for two reasons: (i) it decreases the update magnitude smoothly with increasing uncertainty, thereby dampening instability, and (ii) the denominator guarantees bounded, well-behaved updates without introducing additional hyperparameters. We adopt this simple inverse relation for its ease of implementation, and empirical robustness across tasks without extra tuning.

D. Experimental Setup

TABLE II
EXPERIMENTAL SETUP, NOISE SCHEDULE, AND HYPERPARAMETERS.

Category	Parameters
Training	400 episodes; replay buffer = 100k; ϵ-greedy ($\epsilon_{start} = 1.0$, $\epsilon_{end} = 0.01$).
Noise schedule	No noise: episodes 0–99; burst prob. = 0.01 per step; burst length = 500 steps; cool down = 20k steps; Gaussian noise $\mathcal{N}(0, 1)$.
Hyperparameters	**DQN:** batch=128, lr=0.0005, $\gamma = 0.99$, $\tau = 0.005$, ϵ-decay=0.99. **BootstrapDQN:** batch=64, lr=0.0005, mask prob.=0.9, prior=10, Bernoulli masks. **SunriseDQN:** batch=64, lr=0.0005, mask prob.=0.9, prior=10, ensemble size=5. **IV-DQN:** batch=64, lr=0.0005, $\gamma = 0.99$, $\tau = 0.005$, dynamic ϵ, min batch=48, mask prob.=0.5, ensemble size=5. **Masksemble (Ours):** masks=4, scale=2, hidden layers=[64,64], mask combine=average, lr=0.0005, $\gamma = 0.99$, $\tau = 0.005$, ϵ-decay=0.99, batch=128.

We evaluate our method in both online and offline RL settings, testing robustness under controlled noise in simulations and validating real-world applicability on a large-scale industrial dataset.

Online RL: To destabilize learning, we inject structured observation noise using a `StepBurstNoiseObservation` wrapper. This wrapper is an extension of proposed `RandomNormalNoisyObservation` from [29] that adds Gaussian noise to the observation for multiple steps with a probability p. Unlike prior work [16], [30]–[32], which primarily examines mild or short-lived perturbations, our design introduces sustained and severe noise to deliberately destabilize the agent, thereby testing its ability to maintain stable learning. In the burst-style noise schedule, training begins with 100 warm-up episodes without noise. Afterwards, Gaussian noise $\mathcal{N}(0, 1)$ is injected with probability $p = 0.01$ per step. When triggered, a burst lasts 500 steps (potentially crossing episode boundaries, with episodes capped at 1000 steps) and is followed by a 20,000-step cool-down before another burst can occur. The noise amplitude was fixed at `1.0` across all environments, producing rare but severe perturbations. Even under noise, states are clipped to the valid ranges defined by the Gym environments.

Baselines Reproducibility. This setup creates rare but severe perturbations, in contrast to prior work that typically assumes mild noise, thereby testing whether agents can recover and maintain stability. To validate our framework, we ran experiments on three standard OpenAI Gym environments: *CartPole-v1*, *MountainCar-v0*, and *LunarLander-v2*. The environments were chosen for their simplicity, interpretability, suitability based on previous research [7], [9] for studying noise effects and learning stability.

Offline RL: We also evaluate our method in an offline RL setting using an anonymized real-world customer churn prediction dataset provided by our industrial partner, an insurance company. The dataset consists of over 250,000 policy-level records (e.g., car, fire, theft), each labeled with a binary churn indicator. The data was structured to allow for policy-level prediction of customer churn (policy cancellation). Features broadly include customer demographics (e.g., age and gender, with about 21% of customers aged $\leq$25, 47% between 26–45, and a majority male share of roughly 62%), portfolio composition (types and combinations of subscribed insurance products), policy-specific attributes (such as vehicle age or property characteristics), and behavioral indicators including marketing engagement, quote requests, and claim history. This policy-centric structure, a design choice of the industrial partner, enables the identification of customers at risk of canceling insurance policie(s) they hold with the provider. Features include 24 numerical and 25 categorical variables capturing customer and policy attributes. The data had been preprocessed, cleaned and transformed (dealing with missing values, removing null values, dealing with outliers) by our partner given their domain knowledge yielding a 40-dimensional context representation. At each step, the agent observes a context vector x_t, selects an action $a_t \in \{\text{Churn}, \text{Not Churn}\}$, and receives an immediate reward r_t corresponding to retention or churn. This contextual bandit formulation naturally incorporates the inherent noise present in real-world data. Model performance is evaluated using BA, a popular metric for imbalanced datasets.

Inspired from [23], the dataset was split into 80% training and 20% testing using a random stratified split to preserve class proportions. Each row corresponds to a single policy, and no policy record leaks across splits. Due to Non disclosure agreement restrictions, we cannot disclose the exact churn ratio. However, the proportion of churners is substantially lower than 15%.

In imbalanced datasets, minority samples are harder to classify, so the reward function emphasizes their importance by assigning higher costs. The agent receives $+1/-1$ for correct/incorrect predictions on minority samples and $\lambda/-\lambda$ on majority samples, with $\lambda \in [0, 1]$:

$$R(s_t, a_t, l_t) = \begin{cases} +1, & a_t = l_t \text{ and } s_t \in D_P, \\ -1, & a_t \neq l_t \text{ and } s_t \in D_P, \\ \lambda, & a_t = l_t \text{ and } s_t \in D_N, \\ -\lambda, & a_t \neq l_t \text{ and } s_t \in D_N, \end{cases}$$

where D_P and D_N denote the minority and majority sets, s_t is the state at time t, a_t the agent's action (predicted label), and l_t the true class label. For $\lambda < 1$, minority errors carry higher cost. The best performance is observed when $\lambda = \rho = \frac{|D_P|}{|D_N|}$. When data is balanced, we see that $\lambda = 1$ and costs are equal.

For both online and offline settings, we compare against standard baselines like DQN [28], BootstrapDQN [12] and state-of-the-art UE-based methods SunriseDQN [6], IVDQN [9].

Table II presents the main hyperparameters of the various models. Intuitively, increasing the number of masks in a relatively small network increases the overlap between active

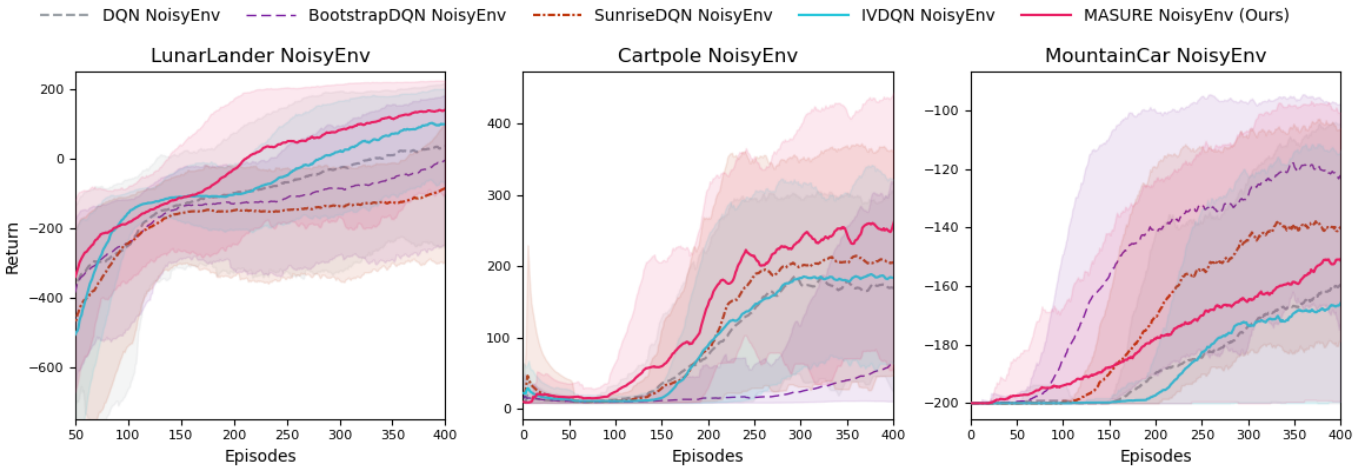

Fig. 1. Comparison of performance of baseline agents and MASURE in noisy environments, showing improved stability of MASURE.

TABLE III
BALANCED ACCURACY (BA) (%) OF RL AGENTS FOR CHURN PREDICTION (OFFLINE LEARNING).

Agent	Balanced Accuracy
DQN	63.52 ± 1.2
BootstrapDQN	63.78 ± 0.9
SunriseDQN	61.88 ± 1.5
IVDQN	62.00 ± 1.3
MASURE (Ours)	$\mathbf{64.30} \pm 1.8$

neurons across subnetworks. This can reduce the effective diversity of the ensemble and may reintroduce the correlation in BootstrapDQN, where multiple heads learn similar representations despite independence.

For online RL benchmarks, performance is reported as the average cumulative reward over the last 50 evaluation episodes, averaged across 8 random seeds. For the offline task, performance is measured using BA on the test set, also averaged across 8 random seeds. All experiments were run on an RTX 3080Ti GPU with an Intel i7 CPU and 16 GB RAM. For more details, please refer to the article and the publicly available code repository of [9].

IV. RESULTS

We evaluate the impact of noisy observations in benchmark environments. As shown in Fig. 2, which compares IVDQN and MASURE under noise-free and noisy settings, the SOTA UE-based IVDQN's performance degrades under burst-style noise injection, whereas MASURE maintains stable learning dynamics, demonstrating robustness to perturbations. In Fig. 1 we present the performance comparison in all 3 environments. MASURE outperforms all baselines in LunarLander and CartPole, providing clear evidence of improved stability under extreme noise.

In the MountainCar NoisyEnv, MASURE exhibits a similar limitation to IVDQN, struggling to exploit exploration effectively. This outcome is expected, as MountainCar provides sparse rewards with limited feedback in most states. The proposed uncertainty-conscious update scales the TD step by $\frac{1}{1+\sigma^2}$, ensuring stability under noisy observations. However, in sparse-reward environments such as MountainCar, states along unexplored trajectories tend to exhibit high epistemic uncertainty but low immediate reward. The resulting reduction in effective learning rate slows value propagation and biases the agent toward familiar states, thereby discouraging exploration. Under these conditions, BootstrapDQN performs best, while IVDQN's weighted strategy does not consistently improve results. Nevertheless, MASURE still achieves better performance than IVDQN, confirming its robustness advantage even in exploration-heavy settings. Overall, while the proposed weighting scheme enhances robustness in noisy, dense-reward tasks, it may underperform in environments where exploration and long-horizon reward propagation are essential for success.

This limitation could be mitigated by coupling the

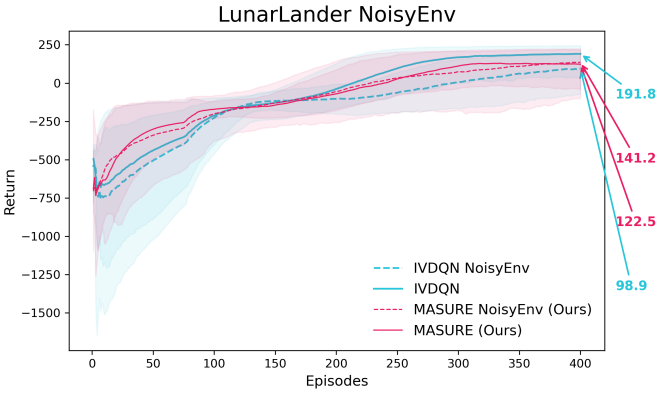

Fig. 2. Comparison of SOTA IVDQN and MASURE in noisy environments, showing improved stability of MASURE.

uncertainty-weighted update with an exploration bonus, similar to the UCB-based mechanism in SunriseDQN [6]. Alternatively, intrinsic motivated approach to stimulate Policy Exploration [33] could be integrated to encourage exploration as well.

For the offline setting, we evaluate on a real-world churn prediction dataset, where we assume inherent noise is present in all features and therefore do not inject additional perturbations. As summarized in Table III, MASURE achieves the highest BA (64.30%), outperforming DQN (63.52%), BootstrapDQN (63.78%), SunriseDQN (61.88%), and IVDQN (62.00%). MASURE significantly enhances stability in NoisyEnv and generalizes effectively to the inherently noisy real-world offline data. Thus, MASURE provides a stable and accurate approach for UE offline learning on real-world datasets.

V. Conclusion

This work introduced MASURE, a Masksembles-based architecture for UE-based RL. Our results demonstrate that weighing updates with uncertainty provides clear added value: the proposed method not only stabilizes training under burst-style noise but also achieves superior predictive performance on a real-world churn prediction task. Compared to standard RL agents and state-of-the-art UE-based baselines such as IVDQN [9], MASURE consistently delivers improved stability and robustness in both offline and online settings.

In simulated OpenAI Gym environments, baseline agents suffered substantial drop in performance when trained with observation noise, while MASURE maintained stable learning dynamics and achieved higher cumulative rewards. In the offline churn prediction task, MASURE attained the best BA, highlighting its practical value in industrial applications where feature noise is inherent. Importantly, by reducing the number of networks required relative to deep ensembles, MASURE addresses the significant computational overhead typically associated with ensemble-based UE. The core contribution of MASURE is the formalization of *uncertainty-conscious Q-value updates* via variance across Masksemble heads, ensuring stable learning even under consecutive noisy states. Although we employ a Q-learning backbone for MASURE, the same principle of uncertainty-weighted updates can be extended to actor–critic methods such as SAC-N [14], where uncertainty is estimated from an ensemble.

Overall, the contributions of this work are threefold: (i) improved predictive performance in real-world offline environments, (ii) enhanced robustness in noisy online settings, and (iii) more efficient uncertainty quantification without the heavy compute of deep ensembles. By reducing the number of networks required relative to deep ensembles, MASURE directly addresses the substantial computational overhead typically associated with ensemble-based UE. As shown by the authors in [8], Masksembles deliver ensemble-level performance and UE at nearly the cost of a single network, avoiding the need to train and evaluate multiple independent models.

A key challenge remains to improve exploration in sparse-reward environments, where both MASURE and IVDQN underperform. Future work could also extend MASURE by explicitly disentangling aleatoric and epistemic uncertainty, allowing agents to treat state noise and model uncertainty differently. Scaling MASURE to high-dimensional continuous-control tasks (e.g., MuJoCo) would provide further evidence of robustness in more complex domains. An ablation study quantifying the individual contribution of the Masksembles architecture and the proposed uncertainty-weighted update rule, as well as the effect of the number of masks, is currently in progress and will be included in an extended version of this work.

Acknowledgment

This research was made possible thanks to the support of Ethias through the HEC Digital Labs.

We would like to thank PyTorch contributors and community [34] for tremendously helpful documentation and code. We also thank Weights and biases [35] for free academic account to log and visualize the training.

References

[1] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, p. 120495, 2023.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[4] Z. Gu, Z. Jia, and H. Choset, "Adversary a3c for robust reinforcement learning," *arXiv preprint arXiv:1912.00330*, 2019.

[5] O. Lockwood and M. Si, "A review of uncertainty for deep reinforcement learning," in *Proceedings of the Eighteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, ser. AIIDE'22. AAAI Press, 2022.

[6] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, "Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning," in *International conference on machine learning*. PMLR, 2021, pp. 6131–6141.

[7] B. Charpentier, R. Senanayake, M. Kochenderfer, and S. Günnemann, "Disentangling epistemic and aleatoric uncertainty in reinforcement learning," *arXiv preprint arXiv:2206.01558*, 2022.

[8] N. Durasov, T. Bagautdinov, P. Baque, and P. Fua, "Masksembles for uncertainty estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 539–13 548.

[9] V. Mai, K. Mani, and L. Paull, "Sample efficient deep reinforcement learning via uncertainty estimation," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=vrW3tvDfOJQ

[10] S. Gadgil, Y. Xin, and C. Xu, "Solving the lunar lander problem under uncertainty using reinforcement learning," in *2020 SoutheastCon*, vol. 2. IEEE, 2020, pp. 1–8.

[11] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving pilco with bayesian neural network dynamics models," in *Data-efficient machine learning workshop, ICML*, vol. 4, no. 34, 2016, p. 25.

[12] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," *Advances in neural information processing systems*, vol. 29, 2016.

[13] Q. Liu, Y. Li, Y. Liu, M. Chen, S. Lv, and Y. Xu, "Exploration via distributional reinforcement learning with epistemic and aleatoric uncertainty estimation," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2256–2261.

[14] G. An, S. Moon, J.-H. Kim, and H. O. Song, "Uncertainty-based offline reinforcement learning with diversified q-ensemble," *Advances in neural information processing systems*, vol. 34, pp. 7436–7447, 2021.

[15] A. Mavor-Parker, K. Young, C. Barry, and L. Griffin, "How to stay curious while avoiding noisy tvs using aleatoric uncertainty estimation," in *International conference on machine learning*. PMLR, 2022, pp. 15 220–15 240.

[16] R. Fox, A. Pakman, and N. Tishby, "Taming the noise in reinforcement learning via soft updates," in *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, ser. UAI'16. Arlington, Virginia, USA: AUAI Press, 2016, p. 202–211.

[17] M. Fortunato, M. G. Azar, B. Piot, J. Menick, M. Hessel, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rywHCPkAW

[18] Y. Flet-Berliac and P. Preux, "Only relevant information matters: Filtering out noisy samples to boost rl," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 2711–2717, main track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/376

[19] A. Li, Z. Chen, T. Klassen, P. Vaezipoor, R. Toro Icarte, and S. McIlraith, "Reward machines for deep rl in noisy and uncertain environments," *Advances in Neural Information Processing Systems*, vol. 37, pp. 110 341–110 368, 2024.

[20] R. Sachdeva, R. Gakhar, S. Awasthi, K. Singh, A. Pandey, and A. S. Parihar, "Uncertainty and noise aware decision making for autonomous vehicles-a bayesian approach," *IEEE Transactions on Vehicular Technology*, 2024.

[21] Y. Zhu, J. Dong, and H. Lam, "Uncertainty quantification and exploration for reinforcement learning," *Operations Research*, vol. 72, no. 4, pp. 1689–1709, 2024.

[22] C. E. Luis, A. G. Bottero, J. Vinogradska, F. Berkenkamp, and J. Peters, "Model-based uncertainty in value functions," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 8029–8052.

[23] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2488–2502, 2020.

[24] 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.

[25] 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.

[26] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," *Advances in neural information processing systems*, vol. 32, 2019.

[27] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in neural information processing systems*, vol. 31, 2018.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[29] R. Khraishi and R. Okhrati, "Simple noisy environment augmentation for reinforcement learning," *arXiv preprint arXiv:2305.02882*, 2023.

[30] X. Chen, X. Liu, C. Luo, and J. Yin, "Robust multi-agent reinforcement learning for noisy environments," *Peer-to-Peer Networking and Applications*, vol. 15, no. 2, pp. 1045–1056, 2022.

[31] K. Sun, Y. Liu, Y. Zhao, H. Yao, S. JUI, and L. Kong, "Exploring the robustness of distributional reinforcement learning against noisy state observations," 2022. [Online]. Available: https://openreview.net/forum?id=z2zmSDKONK

[32] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," *Advances in neural information processing systems*, vol. 33, pp. 21 024–21 037, 2020.

[33] L. Bagot, K. Mets, and S. Latré, "Learning intrinsically motivated options to stimulate policy exploration," in *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020. [Online]. Available: https://openreview.net/forum?id=Vcf1fDmBYJk

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[35] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/