This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

AREPO: Uncertainty-Aware Robot Ensemble Learning Under Extreme Partial Observability

Yurui Du[1], Louis Hanut[2], Herman Bruyninckx[2], Renaud Detry[1, 2]

***Abstract*—Real-world applications of vision-based robot learning face two major challenges: extreme partial observability and effective simulation-to-reality (sim-to-real) transfer. This paper introduces a robust robot learning framework that enhances uncertainty awareness to address these challenges. We reinterpret variational-autoencoder–based visual reinforcement learning (RL) from an uncertainty-quantification perspective, enabling resilience to high sensory noise and severe visual occlusions—common in industrial robotic tasks. To further improve sim-to-real transfer, we propose an uncertainty-aware ensemble RL algorithm. We validate our methods on a laboratory task designed as a proxy for real-world industrial applications characterized by harsh environments with low visibility and physical occlusions. Both simulation and real-world results demonstrate significant improvements in task accuracy and efficiency over various baselines, highlighting the benefits of uncertainty-aware robot learning for complex operational contexts.**

***Index Terms*—Reinforcement Learning, Transfer Learning, Sensorimotor Learning**

I. Introduction

In vision-based robot learning, using partial and noisy observations for policy learning presents a substantial challenge, as a single observational frame often violates the Markov assumption—i.e., that the current observable state contains all necessary information for future decision-making. To address this, three main solutions have emerged: The first involves using recurrent neural networks (RNNs) to encode entire past trajectories [1], [2]. While RNNs theoretically capture comprehensive historical data, they suffer from slow training times and high computational costs due to the sparse nature of reinforcement learning (RL) losses, making them less viable for industrial applications. The second approach constructs a belief state as a statistical summary of past trajectories [3], [4]. However, this method also incurs high computational costs due to the complexity of continuously updating and tracking these dynamic states, compounded by the demand for intensive sequence modeling. The third approach extracts denoised, compressed latent representations from noisy observations using an autoencoder, effectively revalidating the Markov assumption [5], [6]. This allows for the application of efficient, memoryless RL methods, reducing computational overhead and better suiting real-world robotic applications. These autoencoder-based approaches provide a promising compromise between computational efficiency and the ability to handle complex, noisy data streams.

While end-to-end RL allows agents to learn directly from physical interactions with the environment, industrial robotic tasks often preclude large-scale data collection due to high costs and safety risks. To mitigate this, RL is frequently integrated into a simulation-to-reality (sim-to-real) framework, where policies are first developed in simulation before being transferred to real-world settings.

Sim-to-real policy transfer is well-established in deep reinforcement learning (DRL), but existing methods often overlook the challenges of observational uncertainty. Many approaches assume near-complete observability, rely on precise environmental modeling through domain randomization or physics-based methods [7], [8], or require continuous adaptation with cheap access to the target domain [9], [10]. While effective in controlled environments, these methods may struggle in the presence of extreme noise and partial observability, where occlusions obscure true state changes. To address this, platform-agnostic transfer methods [11] mitigate task-specific noise by decoupling perception from control, while Kalman filtering integrated with DRL [12], [13] improves robustness in dynamic, noisy tasks. However, both approaches still depend on accurate environmental modeling, which is often impractical in industrial settings due to the complexity of real-world dynamics and constraints on large-scale data acquisition.

To address these challenges, we revisit visual DRL methods based on variational autoencoders (VAE), from an uncertainty-quantification perspective. We explore their connection to performance gaps in sim-to-real transfer, and we introduce a novel uncertainty-aware ensemble DRL framework. Our approach enhances decision-making under extreme noise and partial observability while fostering an uncertainty-based collaborative ensemble mechanism. This mechanism aids in transitioning from potentially inaccurate models during training to effective real-world applications. We achieve this through a unique en-

Manuscript received: September, 21, 2024; Revised November, 23, 2024; Accepted February, 24, 2025.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the European Union (robetarme-project.eu). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.



Funded by the European Union

[1]Yurui Du and Renaud Detry are with KU Leuven, Dept. Electrical Engineering, Research unit Processing Speech and Images, B-3000 Leuven, Belgium (e-mail: yurui.du@kuleuven.be; renaud.detry@kuleuven.be).

[2]Louis Hanut, Herman Bruyninckx, and Renaud Detry are with KU Leuven, Dept. Mechanical Engineering, Research unit Robotics, Automation and Mechatronics, B-3000 Leuven, Belgium (e-mail: louis.hanut@kuleuven.be; herman.bruyninckx@kuleuven.be).

Digital Object Identifier (DOI): see top of this page.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

semble learning framework that minimizes deviations among individual policies, encouraging emergent behavior aligned with collective wisdom: an uncertainty-weighted sum of all policies within the ensemble, prioritizing the policy with the least sim-to-real gap.

Our contributions are threefold: (1) We propose uncertainty-aware DRL algorithms based on a reinterpretation of VAE-based visual DRL, leading to improved sampling efficiency (Section IV-A) and enhanced sim-to-real transfer performance (Section IV-B). (2) To validate our approach, we design a laboratory task representative of a wide range of real-world applications, including shotcreting [14], sandblasting [15], and paint spraying [16] (Section V). (3) We provide a comprehensive survey and reinterpretation of state-of-the-art VAE-based visual DRL from an uncertainty-quantification perspective, addressing the challenges of extreme partial observability and harsh industrial environments (Section IV-A). Our results demonstrate that policies learned in simulation not only generalize effectively to real-world conditions in a zero-shot manner but also outperform traditional model-based planners [17] and DRL baselines [6], [18]–[20].

II. Related Work

Uncertainty quantification in DRL is critical for providing agents with deeper insights into the learning process. Uncertainty can be categorized into two types: aleatoric and epistemic. Aleatoric uncertainty is inherent to stochastic data and irreducible, whereas epistemic uncertainty arises from the agent's incomplete understanding of the environment. Most uncertainty quantification methods focus on techniques such as bootstrapping and Monte Carlo (MC) dropout.

Bootstrapped Deep Q-Networks (DQN) [21] introduced epistemic uncertainty quantification through a shared network with multiple heads, where variance in head predictions reflects uncertainty. Extensions penalize highly uncertain states via bootstrapped prior Q-networks [22], [23], reweight Bellman backups for exploration in ensemble frameworks like SUNRISE [24], or estimate epistemic uncertainty using cross-entropy between synthetic and true samples, as in SUMO [25]. While SUMO provides robust estimates, its search-based design incurs high computational costs, limiting real-time applicability.

MC Dropout [26] approximates Bayesian inference through multiple stochastic forward passes. Though effective in supervised learning, its uncertainty quality underperforms variational inference methods [27]. In continuous control tasks, combining MC Dropout with bootstrapped Q-values improves uncertainty estimates [28], but repeated inference introduces scalability challenges for real-time robotics. Hybrid approaches [29] integrating MC Dropout and bootstrapped ensembles disentangle aleatoric and epistemic uncertainties, leading to improved decision-making.

Despite these advancements, the role of uncertainty quantification in sim-to-real transfer remains underexplored. Existing methods emphasize theoretical benchmarks while overlooking partial observability and domain shifts common in real-world robotics. This work bridges that gap by leveraging variational uncertainty quantification and ensemble learning to enhance sim-to-real transfer, addressing epistemic uncertainty in dynamic, noisy environments without the inefficiencies of MC Dropout or search-based methods.

III. Preliminaries

In RL, the problem setting is often formulated as an MDP described by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the state transition probability function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and γ is the discount factor.

The policy π_θ is optimized to maximize the cumulative reward $J(\theta)$ of the trajectory $\tau = (s_t, a_t, r_t, ..., s_T, a_T, r_T)$. The cumulative reward is written as

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[R(\tau)], \tag{1}$$

where $R(\tau) = \sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})$. The policy is generally optimized via a gradient-based method, with $\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(a_t|s_t) \cdot A^{\pi_\theta}(s_t, a_t)]$. The advantage function $A^{\pi_\theta}(s_t, a_t)$ is defined as $A^{\pi_\theta}(s_t, a_t) = Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[R(\tau)|s_t, a_t] - \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[R(\tau)|s_t]$.

Generalized advantage estimation (GAE) is widely used to balance bias and variance, stabilizing training [30]. GAE approximates $A^{\pi_\theta}(s_t, a_t)$ as $\hat{A}^{\pi_\theta}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{T} (\gamma\lambda)^{t'-t} [r(s_{t'}, a_{t'}) + \gamma V_\phi(s_{t'+1}) - V_\phi(s_{t'})]$, where $\lambda \in [0, 1]$ is the GAE coefficient, and V_ϕ is the value function estimator learned by minimizing:

$$\mathcal{L}_V(\phi) = \mathbb{E}_{\tau\sim\pi_\theta}[V_\phi(s_{t'}) - R(\tau)]. \tag{2}$$

In this paper, we use proximal policy optimization (PPO) [18], a model-free, on-policy algorithm that constrains policy updates via a clipped surrogate loss:

$$\begin{aligned}\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\min(&w_\theta(s_t, a_t) \cdot \hat{A}^{\pi_\theta}_{\text{GAE}}(s_t, a_t),\\ &\text{clip}(w_\theta(s_t, a_t), 1-\epsilon, 1+\epsilon) \cdot \hat{A}^{\pi_\theta}_{\text{GAE}}(s_t, a_t))],\end{aligned} \tag{3}$$

where $w_\theta(s_t, a_t) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, and ϵ is the clipping factor.

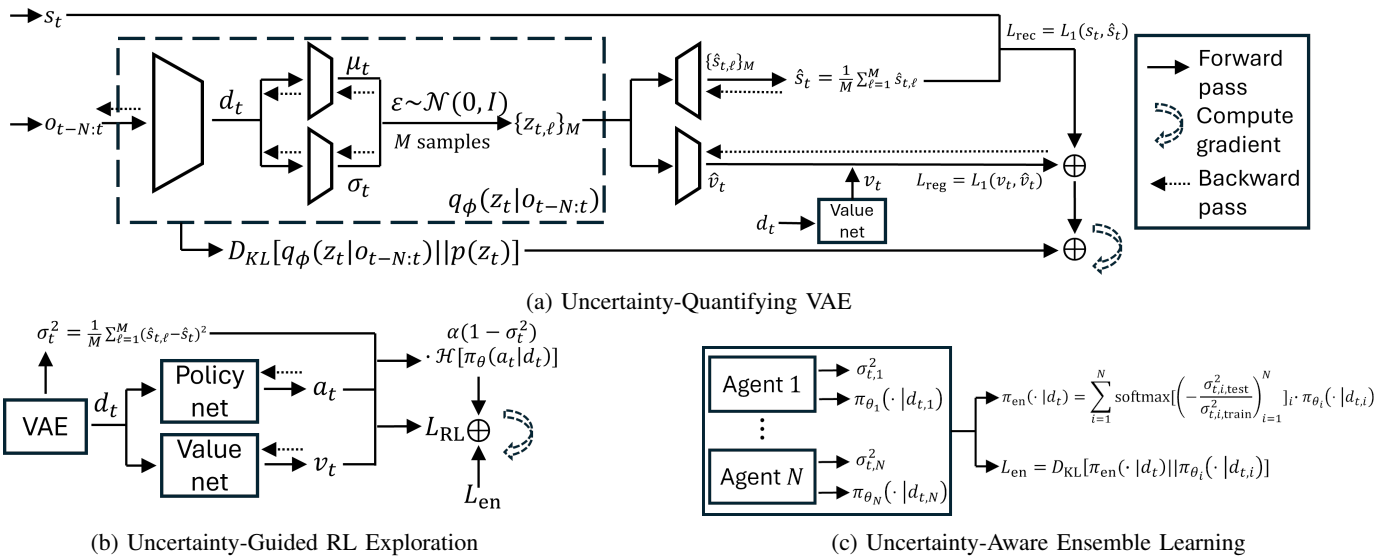
In real-world robotic applications, agents rely on sensors to obtain observations o_t of the state s_t. This complicates RL, as observations are often high-dimensional, noisy, and partially occluded. These challenges have inspired methods that utilize VAEs [5] to learn a denoised, compact latent representation for state inference [31]–[33]. A VAE consists of a convolutional encoder $p_\phi(z_t|o_t)$ that processes noisy, partially occluded observations o_t and generates a latent representation z_t, which serves as input to the policy and value functions instead of s_t. The decoder $p_\varphi(\hat{s}_t|z_t)$, structurally mirroring the encoder, reconstructs an estimate of the corresponding state $\hat{s}_t$ from z_t using transposed convolutional layers that incrementally upscale z_t. The VAE is trained by optimizing the following variational lower bound:

$$\begin{aligned}\mathcal{L}_{\text{VAE}} = &-\mathbb{E}_{z\sim q_\phi(z_t|o_t)}[\log p_\varphi(\hat{s}_t|z_t)] +\\ &\beta \cdot D_{\text{KL}}[q_\phi(z_t|o_t) \,||\, p(z_t)] = \mathcal{L}_{\text{rec}} + \beta \cdot \mathcal{L}_{\text{KL}},\end{aligned} \tag{4}$$

where $q_\phi(z_t|o_t)$ is the variational distribution, and $p(z_t)$ is the Gaussian prior.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451



(a) Uncertainty-Quantifying VAE

(b) Uncertainty-Guided RL Exploration

(c) Uncertainty-Aware Ensemble Learning

Fig. 1: Our algorithm (AREPO) revisits VAE-based RL from an uncertainty quantification perspective. It improves training stability and sampling efficiency with uncertainty-guided exploration and ensemble learning mechanism.

Inspired by these works and empirical evidence from [6], we propose a novel VAE-based DRL network architecture, where a denoised latent representation d_t is learned and passed to the policy and value networks instead of z_t. In the next section, we motivate our design choice.

IV. Method

In this section, we introduce AREPO (Uncertainty **A**ware **R**obot **E**nsemble Learning under Extreme **P**artial **O**bservability). Our approach integrates a denoising VAE for uncertainty-aware state reconstruction, an uncertainty-guided RL strategy that prioritizes exploration based on uncertainty, and an ensemble method that leverages uncertainty awareness for more effective sim-to-real transfer.

A. Reframing VAE-based Visual RL with Uncertainty Quantification

Given a sequence of past observations $o_{t-N:t}$, our goal is to learn $p(d_t|o_{t-N:t})$, from which we can extract a statistically equivalent denoised embedding d_t. This embedding is theoretically sufficient for optimal decision-making because $p(d_t|o_{t-N:t})$ implicitly contains signals of transition dynamics, agent policy, and value functions:

$$\begin{aligned} p(d_t|o_{t-N:t}) &\propto p(o_t|d_t) \cdot p(d_t|o_{t-N:t-1}) \cdot p(o_{t-N:t-1}) \\ &\propto \underbrace{p(d_{t-N}|o_{t-N})}_{\text{encoder}} \prod_{t'=t-N+1}^{t'=t} \underbrace{p(o_{t'}|d_{t'})}_{\text{decoder}} \cdot \underbrace{p(d_{t'}|d_{t'-1}, a_{t'-1})}_{\text{dynamics}} \cdot \\ &\quad \underbrace{\pi(a_{t'-1}|d_{t'-1})}_{\text{policy}} \cdot \underbrace{q(v_{t'-1}|d_{t'-1})}_{\text{value}}. \end{aligned} \tag{5}$$

The observation above is corroborated by positive results from VAE-based DRL [31]–[33]: grounding RL in an autoencoder's latent representation, which implicitly encodes the policy, value, and dynamics, has been shown to improve sampling efficiency and training stability.

However, despite these benefits, previous works based on this method often suffer from suboptimal performance in certain scenarios. Multiple empirical workarounds have been proposed in the literature to mitigate this issue. These include alternating the training of the VAE and RL components [32], [33], using smaller β values in $\mathcal{L}_{\text{VAE}}$, or preventing the policy's gradient from updating other networks except itself [6]. In this section, we revisit VAE-based DRL from an uncertainty quantification perspective, hypothesize key sources of suboptimality in previous work, and propose an efficient and robust uncertainty-guided RL method to address them in a principled manner.

Our objective is to guide RL exploration based on the epistemic uncertainty $p(d_t|o_{t-N:t})$ of the neural network that computes d_t from $o_{t-N:t}$. We measure this uncertainty using the variance of $p(\hat{s}_t|o_{t-N:t})$, as it integrates the uncertainty of $p(d_t|o_{t-N:t})$ when reconstructing the state $\hat{s}_t$:

$$\begin{aligned} p(\hat{s}_t|o_{t-N:t}) &= \int p(\hat{s}_t|z_t) \cdot p(z_t|o_{t-N:t}) \mathrm{d}z_t \\ &= \iint p(\hat{s}_t|z_t) p(d_t|o_{t-N:t}) \cdot \mathcal{N}(\mu_t(d_t), \sigma_t(d_t)) \mathrm{d}d_t \mathrm{d}z_t. \end{aligned} \tag{6}$$

To compute uncertainty tractably, we follow [27] and use Monte Carlo integration to approximate the expectation and variance of the output $\hat{s}_t$ by drawing M samples of $\{z_{t,\ell}\}_{\ell=1}^{M}$ and obtaining M corresponding outputs $\{\hat{s}_{t,\ell}\}_{\ell=1}^{M}$. We use the empirical variance σ_t^2 as the uncertainty measure:

$$\begin{aligned} \sigma_t^2 &= \int (\hat{s}_t - \mathbb{E}[\hat{s}_t])(\hat{s}_t - \mathbb{E}[\hat{s}_t])^T p(\hat{s}_t|o_{t-N:t}) \mathrm{d}\hat{s}_t \\ &\approx \frac{1}{M} \sum_{\ell=1}^{M} (\hat{s}_{t,\ell} - \mathbb{E}[\hat{s}_t])^2, \end{aligned} \tag{7}$$

where $\mathbb{E}[\hat{s}_t] = \int \hat{s}_t \cdot p(\hat{s}_t|o_{t-N:t}) \mathrm{d}\hat{s}_t \approx \frac{1}{M}\sum_{\ell=1}^{M} \hat{s}_{t,\ell}$.

As shown in Fig. 1a, our method differs from previous works in several key aspects:

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

1) We ground the value and policy networks in d_t instead of z_t. The reason behind this design choice is that we model d_t as the denoised latent state that is statistically equivalent to $o_{t-N:t}$, whereas z_t is intended solely for epistemic uncertainty quantification of $p(d_t|o_{t-N:t})$. Therefore, z_t contains undesirable stochasticity and noise for the RL components compared to d_t, leading to subpar performance. This reasoning aligns with the empirical findings in [6], where a diminishing β improves performance and stabilizes training.

2) We integrate training schemes from [32], [33] and [6] by combining alternating and joint optimization strategies. [32], [33] stabilize learning by decoupling VAE and RL optimization, while [6] jointly leverages critic gradients and VAE losses to regulate z_t. To merge these approaches, we introduce a value head $p(\hat{v}_t|z_t)$ parallel to the VAE decoder, mimicking the critic's role in guiding representation learning. This design retains the stability of alternating training while benefiting from critic-driven regularization.

3) Our method uses σ_t^2 to dynamically adjust the maximum entropy objective first proposed in [34] to effectively enhance sampling efficiency. Equation (1) is replaced with an altered maximum entropy RL objective:

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'}) + \alpha_1(1-\sigma_t^2)\mathcal{H}(\pi(\cdot|d_t))], \quad 0 < \sigma_t^2 < 1. \quad (8)$$

The entropy coefficient in Equation (8) serves two key purposes. First, it draws inspiration from uncertainty-driven exploration methods such as MOPO [22] and SUMO [25], which penalize rewards in high-uncertainty regions of learned transition dynamics. Although our approach is model-free and does not explicitly learn dynamics, its uncertainty estimator implicitly captures all RL components, as shown in Equations (5-7). This ensures that the performance guarantees from [22] remain valid in our setting, maintaining the benefits of uncertainty-aware exploration. Second, we extend the original maximum entropy objective [34] by introducing a dynamic adjustment mechanism. Instead of a fixed entropy weight, our approach applies a soft penalty that discourages policy distributions failing to reduce the agent's epistemic uncertainty about the environment. This ensures a principled balance between exploration and exploitation. In Section V-B, we empirically demonstrate that this adaptive entropy regulation significantly improves sample efficiency compared to methods without uncertainty guidance.

B. Approaching Zero-Shot Policy Transfer with Uncertainty Aware Ensemble Learning

Assuming that we can sample N MDPs during training and denote the ith MDP as M_i, our goal is to train a policy π that achieves the best expected test-time reward. Rather than learning a globally optimal policy as a generic POMDP solver might, recent advances in DRL often reduce the global policy learning problem into a set of local policy learning problems to scalably solve MDPs. One such example is LEEP [20], which addresses RL generalization by enabling independent learning

Algorithm 1 AREPO (*training*)

Initialize N training environments with customized parameterizations, initialize policy θ_i, value η_i, and VAE network ϕ_i for each indexed environment $E_i, i \in N$.
for each iteration **do**
  for $i \in N$ **do**
    Collect data $\{s, o, a, s', o'\}_D$ using π_i in E_i.
    for each mini-batch B in D **do**
      Similar to [32] [33], we train the VAE and RL parts in an alternating fashion to stablize training. Activate VAE network ϕ_i.
      for $j \in N, j \neq i$ **do**
        Freeze all networks of agent j, i.e., π_j, η_j, ϕ_j, so agent j is not updated.
      end for
      Compute and update VAE network ϕ_i with $\mathcal{L}_{\text{VAE}}$ computed with Equation (4) and the regularization loss $\mathcal{L}_{\text{reg}} = L_1(v_t, \hat{v}_t)$ in Fig. 1a.
      Freeze VAE network ϕ_i.
      Activate policy and value networks θ_i and η_i.
      Use Equation (3) to compute $\mathcal{L}_{\text{PPO}}$ with altered entropy term in Equation (8) and compute $\mathcal{L}_V$ with Equation (2). $\mathcal{L}_{\text{RL}}$ in Fig. 1b is computed using $\mathcal{L}_{\text{PPO}} + \mathcal{L}_V$.
      Compute $\sigma^2_{t,i,\text{train}} = \mathbb{E}_{\sigma^2_{t,i}\sim S_{i,\text{train}}}[\sigma^2_{t,i}]$, where $S_{i,\text{train}}$ is the set of all computed $\sigma^2_{t,i}$ during training.
      Compute uncertainty measure $\sigma^2_{t,i}$ with Equation (7) as $\sigma^2_{t,i,\text{test}}$. Compute δ_i with Equation (9).
      Compute combined ensemble policy π_{en} with Equation (10).
      Compute ensemble loss $\mathcal{L}_{\text{en}}$ with Equation (11).
      Update θ_i, η_i using $\mathcal{L} = \mathcal{L}_{\text{RL}} + \mathcal{L}_{\text{en}}$.
      Freeze policy and value networks θ_i and η_i.
    end for
  end for
end for

Algorithm 2 AREPO (*inference*)

Load N trained agents, initialize policy θ_i, value η_i, VAE network ϕ_i, and epistemic training uncertainty $\sigma^2_{t,i,\text{train}}$ for each agent $i, i \in N$.
while the task is not done **do**
  Receive observation $o_{t-N:t}$.
  for each agent $i \in N$ **do**
    Compute denoised latent state d_{t_i} using VAE network ϕ_i and $o_{t-N:t}$.
    Compute the corresponding epistemic uncertainty $\sigma^2_{t,i}$ with Equation (7) as $\sigma^2_{t,i,\text{test}}$.
    Compute δ_i with Equation (9).
    Compute policy $\pi_i(\cdot|d_{t,i})$ using policy network θ_i and $d_{t,i}$.
  end for
  Compute combined policy π_{en} with Equation (10).
  Return action $a \sim \pi_{\text{en}}$.
end while

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

for each agent and facilitating cross-agent knowledge sharing through a linker function. In LEEP, the optimal global policy π^* is obtained by combining local policies π_i^* optimal in their respective MDPs using a linker function $\pi^* = f(\{\pi_i^*\}_{i\in[N]})$, where $\{\pi_i^*\}_{i\in[N]} = \arg\max_{\pi_1,\dots,\pi_N} \frac{1}{N}\sum_{i=1}^{N} J_{M_i}(\pi_i) - \alpha_2 \cdot \sum_{i=1}^{N} \mathbb{E}_{p^{\pi_i}_{M_i}}\Big[D_{\text{KL}}[\pi_i||f(\{\pi_i\}_{i\in[N]})]\Big]$.

However, LEEP proposes a probability-based linker function that combines policies by selecting the most probable action. This can limit the capacity of ensemble learning, as each agent is likely to prioritize its own policy during training, and the sim-to-real gap is not explicitly optimized.

To address this issue, we derive an uncertainty-based generalization gap and use it to assign weights to each agent's policy contribution in the linker function. This ensures that the combined policy directly optimizes the generalization gap, leading to improved sim-to-real transfer performance.

The generalization gap from [35], [36] can be written as $\mathbb{E}_{\tau,s\sim p^{\pi}_{M_{\text{train}}}}[R(\tau)|s] - \mathbb{E}_{\tau,s\sim p^{\pi}_{M_{\text{test}}}}[R(\tau)|s]$. Recent work has shown that this gap can be measured by the latent representation deviation Δd_t between training and testing environments, i.e., $||d_{t,\text{test}} - d_{t,\text{train}}||$ [37]. However, this metric is impractical in real-world robotic policy transfer because it requires d_t^{train} and d_t^{test} to share the same underlying state s_t, which is often infeasible to obtain due to noisy and partial observations.

To tractably quantify the generalization gap, we relate Δd_t to the deviation in the reconstructed state $\Delta\hat{s}_t$, i.e., $||\hat{s}_{t,\text{test}} - \hat{s}_{t,\text{train}}||$. Following [37], we assume Lipschitz continuity in a set of functions that map $\hat{s}_t$ to d_t with a constant $L > 0$, yielding $\Delta d_t \leq L\Delta\hat{s}_t$. Similar to [22], we assume an admissible error estimator with upper bounds $\sigma^2_{t,\text{train}}$, and $\sigma^2_{t,\text{test}}$ on the reconstruction errors for both training and testing. Applying the triangle inequality, we obtain: $\Delta d_t \leq L(\sigma^2_{t,\text{test}}+\sigma^2_{t,\text{train}})$. We define the generalization gap as a vector $\boldsymbol{\delta} = (\delta_i)_{i=1}^N$, where δ_i denotes the generalization gap of agent i. By normalizing $\Delta d_{t,i}$ by the continuity constant and the estimator error bound, we define:

$$\delta_i = 1 + \frac{\sigma^2_{t,i,\text{test}}}{\sigma^2_{t,i,\text{train}}}. \tag{9}$$

To leverage the derived uncertainty-based generalization gap for ensemble learning, we propose a linker function based on the generalization gap to combine policies as a weighted sum of individual policies, with weights inversely proportional to the generalization gap. This ensures that agents with smaller sim-to-real gaps have a greater influence on the combined policy, while still incorporating knowledge from other agents:

$$\pi_{\text{en}} = \sum_{i=1}^{N} \text{softmax}(-\boldsymbol{\delta})_i \cdot \pi_i(\cdot|d_{t,i}). \tag{10}$$

We further compute the KL-divergence between each agent's policy and the combined policy, serving as an additional loss term during training. This loss encourages coherence across the ensemble, ensuring that individual policies do not diverge significantly from the ensemble consensus:

$$\mathcal{L}_{\text{en}} = D_{\text{KL}}[\pi_{\text{en}}||\pi_i(\cdot|d_{t,i})]. \tag{11}$$

We propose AREPO, a novel ensemble learning architecture that manages individual agent-environment interactions via shared uncertainty quantification in Fig. 1c. Each agent in the ensemble is instantiated with the uncertainty-quantifying VAE and uncertainty-guided RL modules, described (see Fig. 1a and Fig. 1b respectively). We summarize AREPO in Alg. 1 and Alg. 2.

V. Experiments

This section presents the experimental evaluation of AREPO, addressing two primary questions:

1) Can AREPO achieve improved sampling efficiency and stability compared to uncertainty-unaware DRL baselines under extreme partial observability?

2) Can AREPO generalize to novel scenarios characterized by temporally-correlated dynamics and varying levels of spatially-correlated partial observability?

To investigate these questions, we evaluate AREPO on a task that involves guiding a tool to sprinkle material onto a target surface to a specified thickness (See Fig. 4, displayed on page 7 to conveniently appear next to its sibling Fig. 5). This task serves as a proxy for industrial applications like shotcreting [14], sandblasting [15], and paint spraying [16], where robots must operate under severe visual occlusions and sensory noise. By contrast, standard RL benchmarks used in previous works (e.g., Control Suite [2], [6], Mujoco [10], [19], or Atari [3], [24]) fail to capture temporally and spatially correlated occlusions and noise common in real-world industrial settings. The evaluation is conducted both in simulation and on a laboratory testbed. The simulation allows for systematic variation of key parameters, such as noise, occlusion, and material flow rates, providing a comprehensive performance assessment. The laboratory testbed then validates the sim-to-real policy transfer under real-world constraints.

To further illustrate the task's difficulty and relevance, we evaluate a Model Predictive Control (MPC) [17] planner, and a vanilla planner that operates by always guiding the tool to the position with the least material deposition alongside DRL methods. As is shown in Sections V-C and V-D, the performance deteriorates significantly for all methods except AREPO in scenarios with increasing noise and occlusion, highlighting the difficulty of the task and its suitability for evaluating policies in noisy, partially observable environments.

To address **Question 1**, we evaluate ARPO (the non-ensemble version of AREPO) in simulation against several baselines: DrQ-v2 (encoder-only DRL) [19], MaxEntRL (PPO + maximum entropy objective [34]+VAE), CRL+VAE (MaxEntRL + critic-guided representation learning [6]), ARPO-NVH (SAC+VAE + uncertainty-guided exploration), and ARPO (ARPO-NVH + value head).

In the synthetic experiment addressing **Question 1**, we simulate spatially correlated noise, with severe occlusions concentrated near the contact point where the material meets the surface. Portions of the target surface are also obscured, simulating real-world conditions where dust, debris, and airborne particles reduce visibility and complicate task execution. The material flow rate is modeled as temporally correlated and episodic, with adjustable episode durations to reflect fluctuating pressures or discharge rates typical in real-world

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

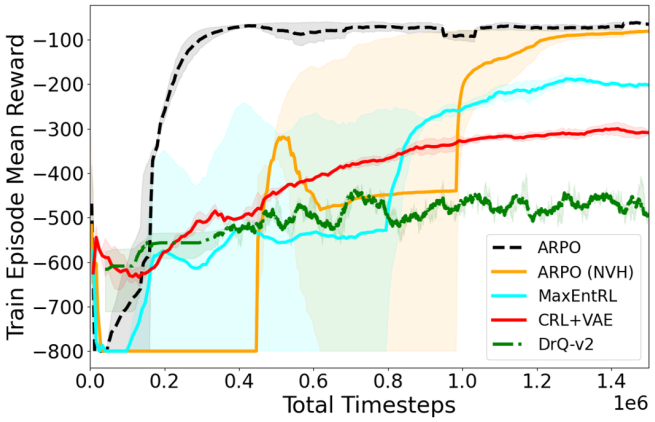


Fig. 2: ARPO achieves the best sampling efficiency and training stability under extreme partial observability compared to DRL methods without uncertainty awareness.

TABLE I: In the experiment on sampling efficiency and training stability, ARPO obtains the best surface quality and most efficient material usage compared to uncertainty-unaware DRL baselines at a slight cost of inference time due to uncertainty quantification. ARPO performs consistently better than MPC and vanilla planners.

Metrics	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)
ARPO	**0.6 ± 0.2**	**3.9 ± 1.4**	**23.8 ± 0.01**	17.2 ± 1.4
CRL+VAE	1.7 ± 0.9	8.3 ± 4.9	32.5 ± 0.02	16.6 ± 2.3
MaxEntRL	1.3 ± 0.6	5.5 ± 2.5	33.8 ± 0.01	**7.7 ± 0.5**
MPC	2.8 ± 0.4	13.5 ± 1.6	31.3 ± 0.05	59.2 ± 2.5
Vanilla	10.2 ± 0.4	40.6 ± 1.5	53.8 ± 0.02	50.1 ± 1.2

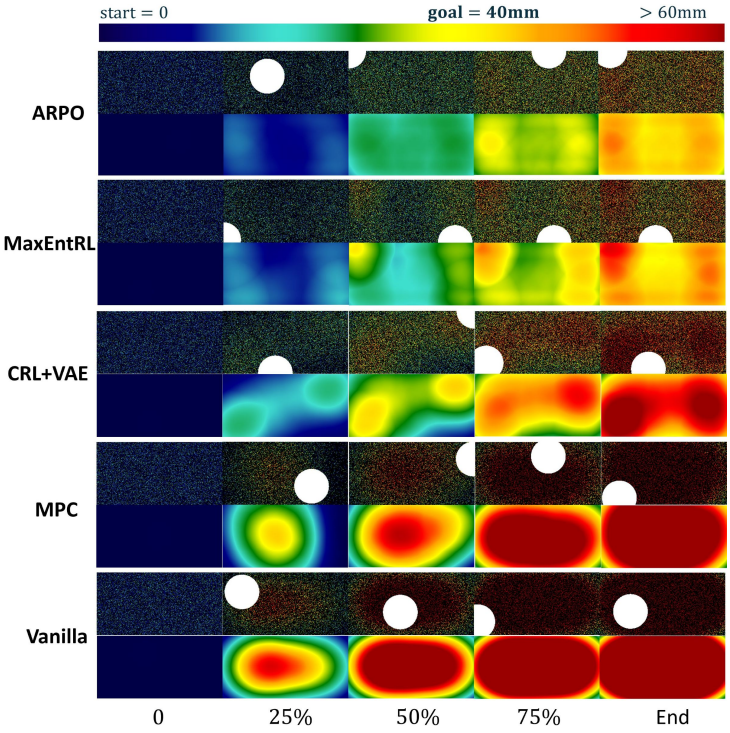


Fig. 3: For each method, the first row displays the observed heightmaps used as input to the models, where the white areas are unobservable to agents. The second row shows corresponding states. The color shift in the heightmaps indicates task progress. ARPO achieves the most uniform surface.

TABLE II: In the experiment on zero-shot sim-to-sim transfer, AREPO remains robust under varying levels of extreme partial observability, achieving better surface quality and robustness than LEEP due to the uncertainty weighted ensemble policy. DR generally achieves the worst performance, indicating a failed sim-to-sim transfer to tasks under extreme partial observability.

Scenarios	Scenario 1: 65% Random Occlusion + 15% Plume				Scenario 2: 80% Random Occlusion + 50% Plume				Scenario 3: 95% Random Occlusion + 85% Plume			
Metrics	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)
AREPO	**1.6 ± 0.3**	**8.1 ± 1.3**	30.0 ± 0.01	32.3 ± 2.2	**1.7 ± 0.5**	**8.6 ± 1.9**	30.0 ± 0.01	32.1 ± 3.2	**1.7 ± 0.3**	**8.8 ± 1.4**	**31.3 ± 0.01**	30.7 ± 2.7
LEEP	3.6 ± 0.6	16.6 ± 2.8	42.5 ± 0.03	28.1 ± 3.6	3.9 ± 0.6	17.9 ± 2.5	40.2 ± 0.03	28.9 ± 3.1	4.6 ± 0.5	20.7 ± 2.0	47.5 ± 0.02	27.2 ± 2.5
DR	4.3 ± 2.6	18.4 ± 4.0	58.8 ± 0.07	**17.5 ± 1.2**	4.5 ± 3.4	18.5 ± 14.0	62.5 ± 0.06	**17.4 ± 1.3**	6.0 ± 3.4	25.7 ± 10.1	71.3 ± 0.01	**17.3 ± 1.5**
MPC	2.8 ± 0.4	13.5 ± 1.6	31.3 ± 0.05	59.2 ± 2.5	3.1 ± 0.5	13.6 ± 1.9	32.5 ± 0.04	59.3 ± 3.2	3.9 ± 0.5	17.1 ± 1.7	37.5 ± 0.03	59.8 ± 2.4
Vanilla	8.9 ± 0.6	35.7 ± 2.36	48.8 ± 0.04	50.2 ± 1.2	10.1 ± 0.7	40.3 ± 2.5	52.5 ± 0.03	50.1 ± 1.2	10.2 ± 0.4	40.6 ± 1.5	53.8 ± 0.02	50.1 ± 1.2

TABLE III: In zero-shot sim-to-real transfer to the laboratory testbed, AREPO achieves the best surface quality compared to baselines without uncertainty awareness.

Metrics	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)
AREPO	**1.1 ± 0.5**	**7.6 ± 1.5**	**24.2 ± 2.1**	30.2 ± 3.2
LEEP	3.3 ± 0.6	17.1 ± 2.8	30.2 ± 2.5	29.2 ± 2.1
DR	4.3 ± 1.8	21.2 ± 4.3	45.7 ± 4.3	**16.7 ± 1.5**
MPC	3.7 ± 0.3	18.7 ± 2.2	32.4 ± 3.1	61.1 ± 3.7

operations. Fig. 3 shows a visual representation of the data generated by our simulator.

To address **Question 2**, We evaluate MPC, DR (vanilla domain randomization) [7], and LEEP (ensemble-based transfer learning method) [20] against AREPO (LEEP + uncertainty-weighted linker function in Equation 9). All DRL methods use ARPO as their agents for a fair comparison.

This experiment consists of two training scenarios, both carried out in simulation: one with a constant material flow rate and another with a extremely varied flow rate ranging from 10% to 1000% of the first rate. Both scenarios feature the same spatially correlated partial observability as in the previous experiment.

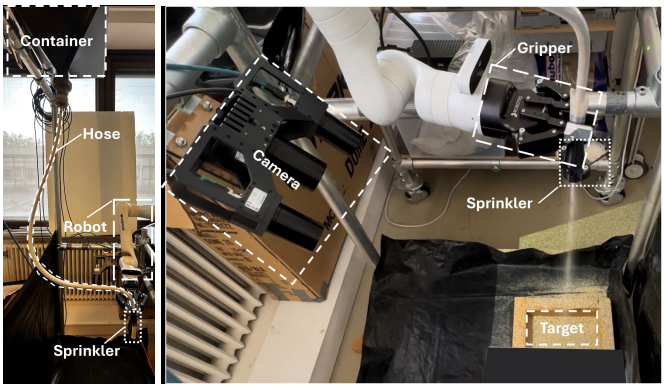
After training, the evaluation of **Question 2** is conducted in two phases. In the first phase, zero-shot tests are performed in simulation under varying degrees of extreme occlusion. This phase evaluates how AREPO and the baseline models adapt to severe partial observability in a controlled simulation. In the second phase, the learned policies are transferred zero-shot to the laboratory testbed. For a detailed layout of the laboratory setup, see Fig. 4a and Fig. 4b.

A. Technical Details

All models in the experiment receive input in the form of 2D heightmaps, representing the material deposition status on the target surface. The output of each model is a 2D velocity vector that controls the nozzle's position in a plane parallel to the target, determining where material is applied.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451



(a) Side view (b) Top view

Fig. 4: Our *sand sprinkling* testbed serves as a proxy for a variety of industrial applications characterized by significant visual occlusions and noise, like for instance tasks that involve spraying granular material. Fig. 4a: Sand stored in an overhead container is transmitted through a hose to a sprinkler positioned over a target (a rectangular bin). Sprinkling generates plumes that create visual occlusions. Fig. 4b: The sprinkler is held by a robotic arm which is guided using heightmaps of the target, derived from images captured by an overhead stereo camera. The heightmaps are illustrated in Fig. 5.

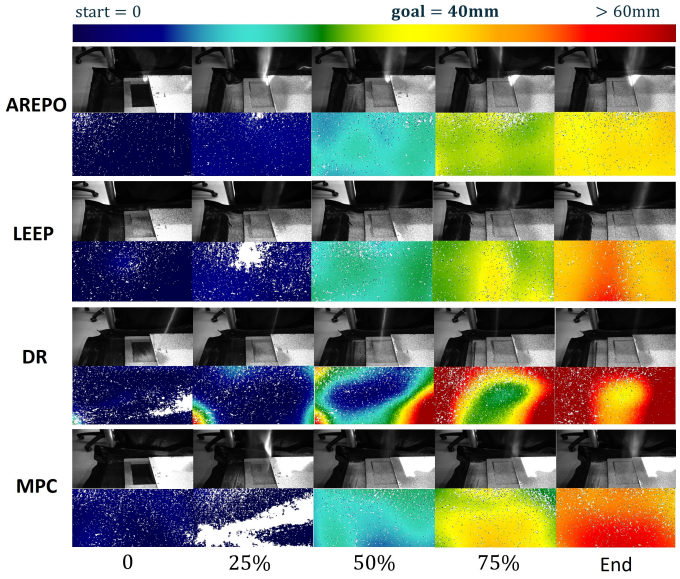


Fig. 5: For each method, the first row displays the grayscale images of the scenario. The second row shows their corresponding heightmaps of the target surface. In the sim-to-real experiment, AREPO is most robust to partial observability caused by dust and sensory noise (unobservable white parts of the heightmaps) and achieves best surface quality compared to other methods.

To account for variability during evaluation, the performance metrics are averaged over 20 independent validation trials. For a fair comparison, all DRL models are implemented using Stable-Baselines3 [38], and trained for 1.6 million timesteps before evaluation. Full details of model architectures and hyperparameters can be found in our publicly available source code at https://gitlab.kuleuven.be/detry-lab/public/arepo.

The performance of the agents are compared with 4 metrics: i) *root-mean-square roughness* $\mathrm{R_{rms}}$: the square root of the mean of the squares of the deviations of the surface height values from the mean surface height, ii) *peak-to-valley roughness* $\mathrm{R_t}$: the difference in height between the highest point and the lowest point on a surface, iii) *waste volume ratio* $\mathrm{r_{wv}}$: the ratio between the wasted volume and the desired volume to be fulfilled. The wasted volume is defined as the material volume that has been sprayed outside the target surface or that exceeds the target thickness. iv) *average inference time* $\mathrm{t_{avg}}$: the average time the agent takes to compute a_t given $o_{t-N:t}$.

B. Simulated Experiment on Sampling Efficiency and Training Stability

Fig. 2 reveals several key insights. First, VAE-based DRL enables more robust representation learning as compared to DrQ-v2 that relies on an encoder-only DRL. Second, SAC+VAE, which augments learning with critic loss, is more stable than MaxEntRL but is more prone to local optimality. Finally, while ARPO-NVH outperforms SAC+VAE by avoiding local optimality using uncertainty-guided exploration, its stability is lower. The inclusion of a value head in ARPO stabilizes training further, highlighting the importance of the value head in balancing exploration and stability. Additionally, Fig. 3 demonstrates ARPO's superior surface quality and minimal material waste, as supported by Tab. I.

C. Simulated Experiment on Zero-Shot Sim-to-Sim Transfer

Tab. II shows that AREPO outperforms LEEP across varying levels of partial observability. This success is attributed to the uncertainty-weighted ensemble policies, which enable AREPO to adapt effectively to occlusion and noise in the environment. In contrast, DR and MPC agents exhibit significant performance deterioration as occlusion increases, indicating failure in sim-to-sim transfer under extreme partial observability. While the vanilla planner shows robust performance due to its simple rules, this simplicity also limits its effectiveness in handling more complex tasks.

D. Real-World Experiment on Zero-Shot Sim-to-Real Transfer

As illustrated in Fig. 5, AREPO demonstrates superior surface quality, achieving a more homogeneous material application across the target surface compared to other methods. Tab. III further quantifies this performance, showing that DR and MPC struggle to generalize effectively, with DR producing the poorest surface quality. Both LEEP and AREPO outperform DR and MPC, highlighting the value of ensemble DRL approaches. AREPO's superiority over LEEP is attributed to its uncertainty-weighted combined policy, which dynamically adjusts based on real-time uncertainty estimations, resulting in more robust material application.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2025.3554451

VI. Conclusions

This paper introduces AREPO, an ensemble DRL framework that enhances robustness under extreme partial observability by incorporating uncertainty quantification through a VAE-based approach, thereby improving sampling efficiency during exploration and training stability via an additional value prediction head. This uncertainty estimation bridges the sim-to-real generalization gap, enabling a more adaptive ensemble mechanism that leverages real-time uncertainty. Empirical results demonstrate AREPO's superior learning efficiency and policy robustness compared to traditional methods. Although this approach increases computational demands, it offers promising research directions in resource-constrained robotic systems and policy transfer across diverse tasks in highly uncertain environments.

References

[1] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 AAAI Fall Symposium Series*, 2015.

[2] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to Control: Learning Behaviors by Latent Imagination,". in *International Conference on Learning Representations*, 2020.

[3] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for pomdps," in *International Conference on Machine Learning*, pp. 2117-2126, 2018.

[4] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal difference variational auto-encoder," *arXiv preprint arXiv:1806.03107*, 2018.

[5] I. Higgins et al., "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR Poster*, vol. 3, 2017.

[6] D. Yarats et al., "Improving sample efficiency in model-free reinforcement learning from images," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021.

[7] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *International Conference on Robotics and Automation*, pp. 3803-3810018, 2018.

[8] F. Djeumou et al. "One Model to Drift Them All: Physics-Informed Conditional Diffusion Model for Driving at the Limits," in *Conference on Robot Learning*, pp. 123–134, 2024.

[9] J. Lyu et al., "Cross-domain policy adaptation by capturing representation mismatch,". in *arXiv preprint arXiv:2405.15369*, 2024.

[10] A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on robot learning*, pp. 262-270, 2017.

[11] D. Li, O. Okhrin, "A platform-agnostic deep reinforcement learning framework for effective Sim2Real transfer towards autonomous driving," in *Nature Communication*, 3, 2024.

[12] H. Lee, M. Kim, S. Park, and D. Kim, "Domain Randomization and Improved Simulation Accuracy for CT Robots," in *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 2345–2352, 2023.

[13] E. Kaufmann, L. Bauersfeld, Loquercio, A. et al. "Champion-level drone racing using deep reinforcement learning," in *Nature*, 620, 2023.

[14] B. Lu, M. Li, T. N. Wong, and S. Qian, "Effect of printing parameters on material distribution in spray-based 3D concrete printing (S-3DCP)," *Automation in Construction*, vol. 124, 2021.

[15] D. J. Taljaard, J. Fourie, and C. P. Kloppers, "The effect of sandblasting and bead blasting on the surface finish of dry electrolyte polishing of laser powder bed fusion parts," *Journal of Materials Engineering and Performance*, vol. 32, pp. 2050-2061, 2023.

[16] G. Tanaka, Y. Takahashi, and H. Iwata, "High precision paint deposition modeling considering variable posture of spray painting robot," in *International Conference on Robotics and Automation*, pp. 2542-2548, 2024.

[17] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed., Springer-Verlag London Limited, 2007.

[18] J. Schulman et al., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[19] D. Yarats et al., "Mastering visual continuous control: Improved data-augmented reinforcement learning," arXiv preprint arXiv:2107.09645, 2021.

[20] D. Ghosh, J. Rahme, A. Kumar, A. Zhang, R. P. Adams, and S. Levine, "Why generalization in RL is difficult: Epistemic pomdps and implicit partial observability," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 25502-25515, 2021.

[21] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped DQN," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[22] T. Yu et al., "Mopo: Model-based offline policy optimization," in *Advances in Neural Information Processing Systems*, 33, pp.14129-14142, 2020.

[23] C. Bai, et al., "Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning," in *International Conference on Learning Representations*, Oct. 2021.

[24] 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*, pp. 6131-6141, 2021.

[25] Z. Qiao, J. Lyu, K. Jiao, Q. Liu, and X. Li. "Sumo: Search-based uncertainty estimation for model-based offline reinforcement learning," in *arXiv preprint arXiv:2408.12970*, 2024.

[26] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, pp. 1050-1059, 2016.

[27] I. Oleksiienko, D. T. Tran, and A. Iosifidis, "Variational neural networks," *Procedia Computer Science*, vol. 222, pp. 104-113, 2023.

[28] T. Hiraoka, T. Imagawa, T. Hashimoto, T. Onishi, and Y. Tsuruoka, "Dropout Q-functions for doubly efficient reinforcement learning," *arXiv preprint arXiv:2110.02034*, 2021.

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

[30] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[31] I. Higgins et al., "Darla: Improving zero-shot transfer in reinforcement learning," in *International Conference on Machine Learning*, PMLR, pp. 1480-1490, 2017.

[32] C. Finn et al., "Learning visual feature spaces for robotic manipulation with deep spatial autoencoders," *arXiv preprint arXiv:1509.06113*, 2015.

[33] A. V. Nair et al., "Visual reinforcement learning with imagined goals," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[34] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence*, 2008.

[35] M. Jiang, E. Grefenstette, and T. Rocktäschel, "Prioritized level replay," in *International Conference on Machine Learning*, pp. 4940-495, 2021.

[36] R. Raileanu and R. Fergus, "Decoupling value and policy for generalization in reinforcement learning," in *International Conference on Machine Learning*, pp. 8787-8798, 2021.

[37] J. Lyu, L. Wan, X. Li, and Z. Lu, "Understanding what affects generalization gap in visual reinforcement learning: Theory and empirical evidence," *arXiv preprint arXiv:2402.02701*, 2024.

[38] A. Raffin et al., "Stable-Baselines3: Reliable reinforcement learning implementations," 2021, [Online]. Available: https://github.com/DLR-RM/stable-baselines3.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on March 28,2025 at 11:24:25 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.