2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Demonstration-Augmented Deep Reinforcement Learning with Mixed Reality Human-in-the-Loop Guidance

Mohammad-Ehsan Matour[1] and Alexander Winkler[1]

Abstract— **The integration of human expertise into reinforcement learning has gained increasing attention as a means to improve sample efficiency and stability. Current approaches often depend on pre-collected expert demonstrations or virtual reality setups, which are costly to generate and difficult to adapt to dynamic training conditions. In this work, a framework is introduced that augments deep reinforcement learning with real-time demonstrations provided through mixed reality interaction. A structured robotic pick-and-place task serves as the benchmark, where a robot must execute sequential phases of grasping, transporting, and releasing an object. Expert guidance is delivered via mixed reality annotations, which are converted into reference trajectories and injected into the learning process whenever performance falls below a predefined threshold. A modified replay buffer accommodates both agent-generated and expert-generated transitions, allowing controlled sampling with a dynamically adjusted expert-to-agent ratio. Training in the real workspace through mixed reality reduces the simulation-to-reality gap considerably, as confirmed by experiments on a physical robot platform. Experimental evaluation demonstrates that the proposed framework accelerates policy convergence, ensures stability under noisy feedback, and achieves strong generalization to unseen task configurations. These findings highlight the potential of demonstration-augmented reinforcement learning through mixed reality as a data-efficient and robust approach to robot training in real-world scenarios.**

I. INTRODUCTION

Deep reinforcement learning (DRL) has become a powerful approach for enabling robots to acquire complex behaviors through trial and error. Unlike traditional programming, DRL allows agents to adapt to dynamic environments and learn directly from interaction, making it suitable for tasks where explicit modeling is difficult or costly [4]. However, applying RL in robotics remains challenging. Training policies directly on physical systems is often impractical due to high sample requirements, high variance in learning outcomes, and the risk of unsafe actions [5]. Although simulation can mitigate these issues, it introduces the sim-to-real gap, where differences between simulated and real-world dynamics can significantly reduce policy performance.

To address these challenges, recent research has explored the integration of human expertise into the learning process [6]. Demonstration-augmented DRL has emerged as a promising direction, which improves sample efficiency, convergence speed, and policy stability. Most existing methods rely on either pre-collected datasets or virtual reality (VR) interfaces. Although these strategies have shown effectiveness, they involve certain limitations. Pre-collected demonstrations are static, task-specific, and cannot adapt to an agent's evolving needs. VR-based methods require specialized hardware and do not guarantee spatial alignment with the robot's workspace. Consequently, the deployment of such approaches in real-world systems remains limited.

Mixed reality (MR) provides a natural alternative by overlaying digital annotations directly into the physical workspace and enables intuitive, spatially aligned human guidance. MR allows experts to provide demonstrations in real time, reducing ambiguity and bridging the gap between simulated training and physical execution. Furthermore, guidance can be delivered selectively when needed, avoiding the inefficiencies associated with fully supervised methods such as behavior cloning (BC) [7], [8], while maintaining the autonomy of RL.

In this work, a framework is proposed that integrates MR-based demonstrations into DRL for structured robotic tasks. A pick-and-place scenario is chosen as a representative benchmark due to its sequential phases. Expert annotations are injected into the training process whenever performance falls below a predefined threshold. The replay buffer is extended to store both agent- and expert-generated transitions, while an adaptive sampling strategy regulates the expert-to-agent ratio over time. This design accelerates learning through guidance while ensuring policies remain robust and do not become overly dependent on demonstrations. Robustness is further analyzed in simulation by injecting Gaussian noise into expert annotations, to simulate the limited accuracy of such inputs. Finally, experiments on a physical robot validate the approach under MR-aligned conditions, demonstrating reduced sim-to-real discrepancy.

The contributions of this work are summarized as follows:

1) Introduction of a demonstration-augmented DRL framework with real-time MR annotations, providing a novel interface for integrating human guidance into structured robotic tasks.
2) Proposal of a modified replay buffer with expert tagging and adaptive sampling, enabling controlled blending of agent and expert experience.
3) Validation of robustness against noisy annotations in simulation, confirming stability under perturbed inputs.
4) Evaluation in both simulation and on a physical robot platform, showing faster convergence, improved generalization to unseen configurations, and enhanced sim-to-real transfer compared to baseline learning.

[1]Mittweida University of Applied Sciences, Mittweida, 09648, Saxony, Germany (matour, alexander.winkler)@hs-mittweida.de

II. Related Works

A. Learning from Demonstration in DRL

Learning from demonstration (LfD) has long served as a foundational strategy in both robotics and DRL. Approaches in this domain include behavior cloning (BC), hybrid pre-training, and replay buffer augmentation. These methods differ in how expert data are collected, integrated into training, and balanced with agent-generated experience.

BC directly trains an agent to imitate expert demonstrations captured in specific contexts. Although straightforward to implement, BC typically exhibits limited generalization to unseen states, as it bypasses direct interaction with the environment [10]. To overcome this, recent studies have combined BC pre-training with RL fine-tuning [11], [12], [13], [3]. Alternative approaches have introduced implicit BC, which reduces sensitivity to deviations from expert behavior and enhances adaptability in dynamic environments [14], [15].

A notable example is the Demonstration-Augmented Policy Gradient (DAPG) method, initially proposed in [3] and later refined in [16]. The updated version decreases dependence on demonstrations while maintaining sample efficiency. Unlike earlier methods focused primarily on simulation, [16] emphasizes real-world training, showing that model-free RL can solve dexterous manipulation tasks without requiring detailed physical models or high-precision simulation environments.

More recent developments have explored multi-modal supervision by incorporating natural language instructions and video demonstrations [17], [18], [19]. These approaches extend beyond state–action pairs, which enables more expressive guidance and improved generalization across a wider range of tasks.

B. Replay Buffer Augmentation and Off-Policy Learning

An alternative research direction focuses on replay buffer augmentation and off-policy learning. In this context, expert demonstrations are integrated into the RL process to accelerate policy optimization while preserving exploratory behavior [20]. Off-policy algorithms such as DDPG make use of replay buffers to store and reuse past experiences, which is particularly advantageous for tasks with sparse rewards.

A notable method in this domain is DDPG from Demonstrations (DDPGfD) [9], where expert demonstrations are preloaded into the replay buffer prior to training and retained alongside agent-generated transitions. The use of prioritized experience replay ensures that expert samples remain influential throughout the training process. Despite its effectiveness, the static nature of demonstrations limits adaptability, posing challenges in real-world scenarios where demonstration data may not be consistently available.

Advantage-Weighted Actor–Critic (AWAC) [21] addresses early-stage learning by combining offline datasets for pre-training with online fine-tuning. While this hybrid approach improves stability during initial training phases, it remains dependent on a fixed dataset and lacks the flexibility to accommodate dynamic expert interventions during learning. As a result, its applicability in interactive or evolving environments is limited.

C. Virtual and Mixed Reality Interfaces for Robot Training

The integration of VR/AR/MR into robot learning has been investigated from several perspectives. Early studies employed AR environments with fiducial markers to train RL agents, but these remained proof-of-concept demonstrations without direct application to robotic training or human-in-the-loop interaction [28]. Mixed-reality architectures have also been proposed for reach-to-grasp tasks, where physical target information is transferred to simulation and learning is performed virtually before execution on the real robot [29].

Later research used AR and MR as perception and safety interfaces in human–robot collaboration. Li et al. [30] proposed an AR-assisted DRL framework where AR devices provide spatial and environmental information, digital twin previews, and collision detection. In follow-up work, Li et al. [31] incorporated MR head-mounted displays (HMD) into a soft actor–critic framework to deliver semantic state information, coordinate transforms, and collision feedback for robot motion planning. In both cases, MR was used primarily as a passive perception medium rather than as an active demonstration interface.

Extensions of this perception-oriented role include a mixed perception-based collaborative maintenance system combining AR-based gesture recognition with DRL decision-making [26], and AR/VR applied in inverse RL for predicting human intent in handover tasks [27]. In these cases, extended reality (XR) technologies were employed to capture or forecast human motion trajectories, facilitating intent recognition rather than shaping robot learning directly.

In contrast, the proposed framework, positions MR as an active demonstration interface. Human-provided corrective actions are injected into the replay buffer with explicit expert tagging, adaptive sampling, and threshold-triggered replacement mechanisms. Unlike the above approaches, which employ XR for visualization, perception, or intent prediction, this framework directly influences the training process by enabling real-time expert guidance during learning.

Prior research has shown that expert knowledge can accelerate policy learning. However, existing approaches face notable limitations. Methods such as BC, replay buffer augmentation, and human-in-the-loop RL often rely on static datasets, specialized hardware, or evaluative feedback rather than structured, real-time demonstrations. Although VR and AR have been used for visualization or perception, MR remains underexplored as an active demonstration interface in RL. This gap is addressed in the present work through a framework that leverages MR for adaptive, real-time demonstrations, which enables efficient learning and improved sim-to-real transfer.

III. Problem Definition and Background

A. Robotic Task Definition

The task under consideration is a pick-and-place operation performed within a structured workspace on a planar surface. A single object is placed within a predefined region, and the robot is required to grasp and transport it to a designated target location. This scenario is intentionally selected due to its structured nature, and allows evaluation of RL in a sequential robotic task with clearly defined objectives. Despite its simplicity, the task reflects key aspects of robotic manipulation, namely approaching, grasping, transporting, and releasing, while remaining representative of industrial applications. To introduce variability and reduce overfitting, the initial object position is randomized within a bounded range across training episodes.

The task is decomposed into five sequential phases: (1) approach, (2) grasp, (3) lift, (4) transport, and (5) release. Progress through these phases provides a natural structure for defining intermediate objectives and for monitoring learning performance. Successful execution is achieved when the object is placed within a tolerance region around the target position. Failure is defined as the inability to grasp the object, the object being dropped during transport, or its release outside the tolerance region. This decomposition supports both the formulation of reward signals and the design of task-progress indicators later used in the learning framework.

B. Reinforcement Learning Framework

The robotic task is modeled as a finite-horizon Markov decision process (MDP) defined by the tuple (S, A, P, R, γ). At each discrete timestep $t \in \{0, \dots, T\}$, the agent observes a state $s_t \in S$, selects an action $a_t \in A$, and receives a reward $r_t = R(s_t, a_t)$, while the environment transitions according to the probability distribution $P(s_{t+1} \mid s_t, a_t)$. The goal is to find a policy $\pi_\theta(a \mid s)$ that maximizes the expected discounted return

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right], \tag{1}$$

where $\gamma \in [0,1]$ is the discount factor. In practice, the policy is implemented as a deterministic actor $\mu_\theta(s_t)$ following the DDPG framework, with exploration noise added during training.

a) State space.: The state vector at timestep t is defined as

$$s_t = \left[p_t^{tcp}, g_t, \phi_t\right], \tag{2}$$

where $p_t^{tcp} \in \mathbb{R}^3$ is the Cartesian position of the end-effector, $g_t \in \{-1, +1\}$ indicates the gripper state (open or closed), and $\phi_t \in [0,1]$ is a scalar task-progress indicator encoding advancement through the sequential phases, that increments when each subtask (approach, grasp, transport, release) is completed. This design provides explicit awareness of task progress, which facilitates learning in long-horizon scenarios.

b) Action space.: The action vector is defined as

$$a_t = \left[p_t^{tcp'}, g_t'\right], \tag{3}$$

where $p_t^{tcp'} \in [-1,1]^3$ represents the normalized target end-effector position in Cartesian space, and $g_t' \in [-1,1]$ is the gripper command. A value $g_t' > 0$ corresponds to closing, while $g_t' < 0$ corresponds to opening. All components are normalized to $[-1,1]$. This absolute parametrization simplifies integration of expert demonstrations, as externally provided poses can be directly injected into the replay buffer without additional transformations.

c) Reward function: The reward function is composed of a dense shaping term, a penalty for premature gripper closure, phase-dependent bonus terms, and additional components that promote time efficiency and proper task termination. At each timestep, the tracking error with respect to the current phase target is

$$d_t = \|p_t^{tcp} - p_{\kappa_t}^{tar}\|_2,$$

where $\kappa_t \in \{0,1,2,3,4\}$ is the phase index. The instantaneous reward is

$$r_t = -\alpha\, d_t \;-\; \eta\, \mathbb{I}(\kappa_t = 0 \wedge g_t = 1) \;+\; r_t^{\text{phase}} \;-\; \lambda \;+\; r_t^{\text{terminal}}, \tag{4}$$

where $\alpha, \eta, \lambda > 0$ are weighting parameters, $g_t \in \{-1,1\}$ encodes the gripper state, and $\mathbb{I}(\cdot)$ denotes the indicator function. The second term penalizes premature gripper closure in the approach phase, and the constant step penalty λ discourages unnecessarily long episodes.

Phase-completion rewards are assigned when the tolerance region of the active target is reached and the corresponding gripper condition is satisfied:

$$r_t^{\text{phase}} = \begin{cases} b_{\kappa_t}, & \text{if } d_t < \varepsilon \text{ and } g_t = \bar{g}_{\kappa_t}, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

with tolerance $\varepsilon > 0$, required gripper states $\bar{g} = [-1,1,1,1,-1]$ for phases $\kappa = 0,1,2,3,4$, and monotonically increasing bonuses $b_\kappa = \kappa + 1$.

Finally, terminal rewards explicitly encode task success or failure:

$$r_t^{\text{terminal}} = \begin{cases} +R_{\text{success}}, & \text{if } \kappa_t = 4,\ d_t < \varepsilon,\ g_t = 0, \\ -R_{\text{fail}}, & \text{if termination without success,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

This formulation combines dense guidance, structured phase-completion feedback, time efficiency, and explicit terminal outcomes. As a result, training is stabilized and the reward remains aligned with physically meaningful task execution.

C. Expert Demonstration Integration

1) Mixed Reality as a Demonstration Interface: MR provides a workspace-aligned mechanism for delivering expert actions during training. At timestep t, an expert annotation is represented as an action $a_t^e \in A$ paired with the observed

state $s_t \in S$. Since the annotations share the same action space as the agent, they are directly compatible with the RL formulation in Section III. Unlike static datasets, MR demonstrations are available on demand and are spatially consistent with the real workspace.

2) Conceptual Integration into Reinforcement Learning: Each transition is extended with an expert indicator:

$$\tau_t = (s_t, a_t, r_t, s_{t+1}, \zeta_t), \tag{7}$$

where $\zeta_t \in \{0,1\}$ specifies whether the action comes from the expert ($\zeta_t = 1$) or from the agent ($\zeta_t = 0$). The replay buffer therefore stores both agent-generated and expert-generated transitions in a unified structure.

The expert policy π_E is deterministic, providing a single action for each state:

$$\pi_E(a_t \mid s_t) = \delta(a_t - a_t^e), \tag{8}$$

where $\delta(\cdot)$ is the Dirac delta distribution centered on the demonstrated action.

During training, minibatches are sampled from the replay buffer according to a mixture distribution over transition types. The probability of sampling an expert transition is controlled by $\lambda_t \in [0,1]$:

$$\mathbb{P}(\zeta_t = 1) = \lambda_t, \quad \mathbb{P}(\zeta_t = 0) = 1 - \lambda_t. \tag{9}$$

This scheduling ensures that early updates are guided primarily by expert demonstrations, while later updates are dominated by agent-generated data, resulting in a gradual transition from guided exploration to independent policy learning (see Figure 9).

Accordingly, the optimization objective extends the standard RL problem to incorporate both data sources:

$$J(\pi_\theta) = \mathbb{E}_{\tau_t \sim \mathscr{D}(\pi_\theta, \pi_E)} \left[\sum_{t=0}^{T} \gamma^t r_t \right], \tag{10}$$

where $\mathscr{D}(\pi_\theta, \pi_E)$ denotes the replay-buffer distribution induced by a mixture of agent and expert transitions. This formulation integrates demonstrations at the data-distribution level while preserving agent autonomy during environment interaction.

IV. Methodology

A. System Overview

The framework consists of three main components: a robot execution system, an MR annotation interface for expert interventions, and a PC-based RL pipeline. Workspace-aligned annotations are streamed to the training process and integrated into the learning loop via the intervention mechanism and bounded projections described in Sections IV-B and IV-C. Because the annotations are expressed in the robot's coordinate frame, consistency between MR input and physical execution is ensured. Figure 1 shows the integrated setup, where holographic markers, coordinate frames, and virtual object representations provide corrective guidance at different phases of the task.

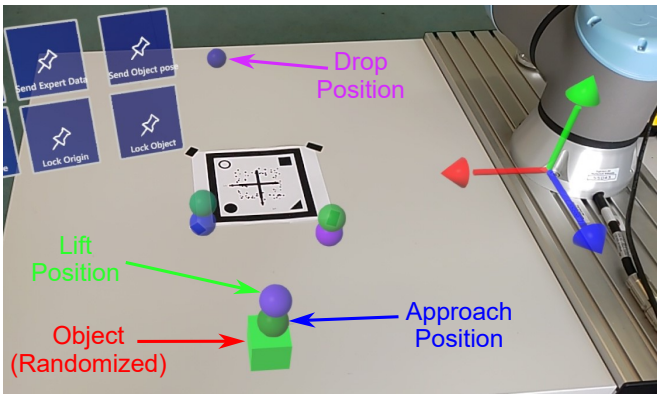

Fig. 1. System overview with MR interface. A Microsoft HoloLens projects holographic annotations into the robot workspace, providing expert demonstrations in a coordinate frame consistent with the robot. The object (red) is randomly placed. Shown are three of the five task phases: approach (blue), lift (green), and drop (magenta).

B. Expert Data Injection Mechanism

Expert demonstrations are integrated at the replay-buffer interface, with intake enabled only during the first 10k steps (half of training), while the remaining steps proceed under autonomous learning. Each transition follows the augmented representation in Eq. 7, where the binary indicator $\zeta_t \in \{0,1\}$ specifies whether the action originates from the expert or from the agent. This structure allows both data sources to be stored in a unified buffer while keeping their origin identifiable for later weighting or scheduling.

1) Replay Buffer Sampling: During mini-batch construction, transitions are sampled according to the probability distribution defined in Eq. 9. The parameter λ_t controls the likelihood of selecting an expert transition, that provides a controlled blend of expert and agent data for updating the policy.

2) Dynamic Expert Ratio Scheduling: The expert ratio λ_t decreases over training to ensure a gradual shift from guided to autonomous learning. A linear schedule is used:

$$\lambda_t = \max\left(\lambda_{\min},\ \lambda_{\max} - \frac{t}{T}\left(\lambda_{\max} - \lambda_{\min}\right)\right), \tag{11}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ denote the initial and final sampling ratios, t is the current training step, and T the training horizon. This schedule prioritizes expert demonstrations during the early phase of training while gradually shifting emphasis toward agent-generated experience.

3) Threshold-Based Activation: Expert guidance is activated if, at any point within an episode, the cumulative reward falls below a threshold R_{th}. Formally, for an episode of length H,

$$\exists t \in \{0, \ldots, H\} : \sum_{k=0}^{t} r_k < R_{\text{th}}. \tag{12}$$

When this condition is satisfied, the suboptimal agent actions of that episode are replaced with the corresponding expert actions a_t^e, and the transition tuple $(s_t, a_t^e, r_t^e, s_{t+1}, \zeta_t = 1)$ is stored in the replay buffer.

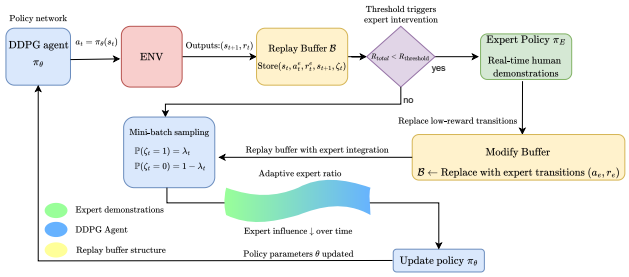

Fig. 2. Flow of the proposed framework. The DDPG agent interacts with the environment, storing transitions in the replay buffer. When cumulative reward falls below the threshold, expert demonstrations a_t^e from the MR interface are injected, replacing low-reward transitions. Mini-batch sampling with an adaptive expert ratio provides a blended buffer for policy updates, with expert influence decaying over time.

C. Stability Measures for Expert Annotations

MR annotations are inherently subject to measurement noise and occasional outliers. To ensure stability during training, expert actions a_t^e are constrained before being stored in the replay buffer. Given the mixed reality pose mr_t, the stored action is defined as

$$\tilde{a}_t^e = a_t^e + \frac{\Delta}{\max(\|mr_t - a_t^e\|_2, \Delta)} (mr_t - a_t^e), \qquad (13)$$

where $\Delta > 0$ sets the maximum allowed deviation. This operation suppresses extreme outliers while keeping the natural variability that is useful for learning.

No artificial noise was introduced during training, since MR input already contains stochastic errors. Robustness to additional Gaussian perturbations is evaluated separately in Section V-E, where controlled noise levels are used to assess the tolerance of the framework.

Figure 2 provides an overview of the overall functionality of the proposed framework. The diagram illustrates how agent interact with the environment, how transitions are stored in the replay buffer, and how expert interventions are injected when performance falls below the threshold. It also highlights the adaptive expert ratio, which regulates the contribution of demonstrations during sampling, and enables a gradual transition from expert-guided to agent-driven learning.

V. Experiments and Results

A. Experimental Setup

Experiments were conducted in both simulation and on a physical robot platform. The policy was trained using DDPG, extended with threshold-based expert activation and replay replacement, an adaptive expert-to-agent sampling ratio, and noise-bounded projection of demonstrations.

Episodes had a finite horizon of T steps, and object positions were randomized to induce variability.

For real-robot validation, the trained policies were deployed on a cobot equipped with a vacuum gripper. MR annotations were provided via a Microsoft HoloLens and calibrated using fiducial marker tracking (Vuforia SDK).

TABLE I

Training parameters used in simulation and real-robot validation.

Replay buffer size	1×10^6
Batch size	256
Learning rate	5×10^{-4}
Discount factor γ	0.99
Expert ratio schedule	$\lambda_t : 0.9 \rightarrow 0.1$ (linear)

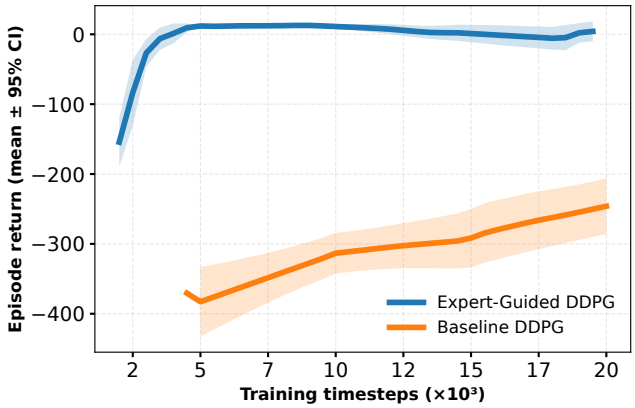

Fig. 3. Learning curves of baseline DDPG and expert-guided DDPG over 20k timesteps. Mean episode return across 10 seeds with 95% confidence intervals.

The main training parameters are summarized in Table I. Experiments were repeated across multiple random seeds, and results are reported as mean values with 95% confidence intervals.

Evaluation metrics included cumulative episode reward (learning curves), convergence speed in terms of training timesteps until stable performance, and task success rate across generalization and sim-to-real evaluations. In addition, robustness was assessed by measuring convergence probability under noisy expert annotations.

B. Learning Performance

Figure 3 compares the training progress of baseline DDPG and the proposed expert-guided variant across ten random seeds. The expert-guided method converged within approximately 5k timesteps, whereas the baseline required significantly more samples and remained far from convergence after 20k timesteps. The narrower confidence intervals highlight the improved stability of the expert-guided approach compared to the large variance of the baseline. These results demonstrate that MR demonstrations considerably accelerate training and reduce variability across seeds.

Figure 4 shows the long-horizon baseline. While the mean return increased slowly, only 2 of 10 seeds converged before 100k timesteps. This highlights the inefficiency of autonomous exploration alone. By contrast, the expert-guided variant reduced sample complexity by more than an order of magnitude while achieving similar final performance.

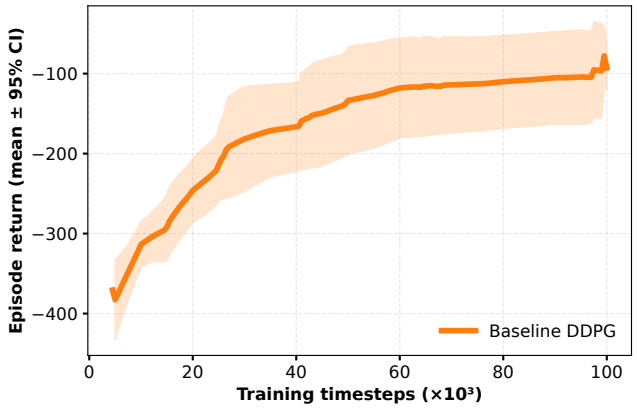

Fig. 4. Long-horizon performance of baseline DDPG up to 100k timesteps. While improvement occurs, convergence is significantly slower than the expert-guided variant.

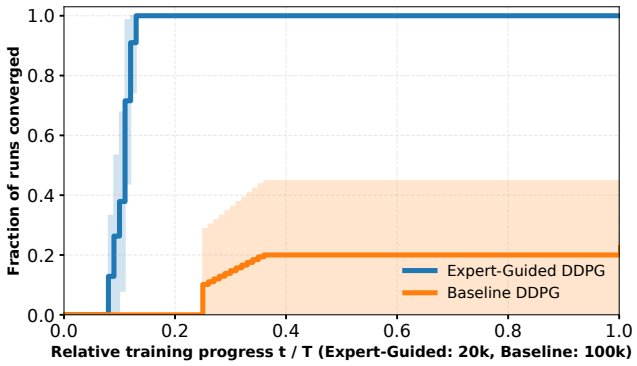

Fig. 5. Convergence probability over normalized training progress. The expert-guided method converged rapidly and consistently, while the baseline showed delayed and unreliable convergence.

C. Convergence Speed

Figure 5 shows the proportion of runs that reached convergence over normalized training progress, where convergence is defined as the point at which the agent's success rate, averaged over the last 20 episodes, stays above the task threshold, which indicates reliable task completion rather than isolated successes. The expert-guided method converged rapidly, with nearly all runs reaching convergence within the first 20% of training. In contrast, the baseline progressed slowly, with only 2 of 10 runs converging within the 100k-step limit. This demonstrates that incorporating expert demonstrations yields much faster convergence while also improving consistency across random seeds.

D. Generalization to Unseen Configurations

To assess spatial generalization, the policy was tested on unseen target positions placed at increasing distances from this region.

Figure 6 reports the mean success rate for groups of targets binned by their nearest distance to the training set. Each target success rate was computed by averaging results over ten seeds with twenty evaluation episodes per seed. Labels

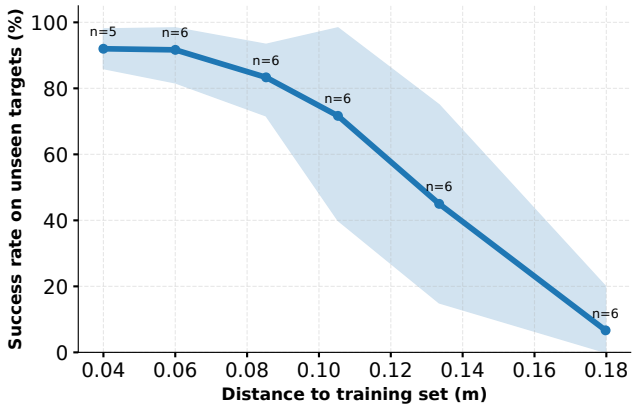

Fig. 6. Generalization performance as a function of distance from the training region. Labels n denote the number of evaluation targets per bin. Shaded regions show 95% confidence intervals. Performance is robust near the training region but degrades at larger distances, reflecting reduced extrapolation capability.

n denote the number of distinct targets included in each distance bin.

Close to the training region, the policy achieved success rates above 90%, confirming reliable transfer to nearby unseen targets. Performance decreased steadily with distance, dropping below 20% at the maximum offset of 0.18 m. The widening confidence intervals indicate greater variability across evaluation targets. The results show that the policy generalizes well to novel but nearby configurations, while its ability to extrapolate far outside the training distribution remains limited.

E. Robustness to Annotation Noise

Robustness to noisy demonstrations was evaluated in simulation by perturbing expert actions with Gaussian noise of standard deviation σ. Figure 7 plots the probability of convergence within the training horizon as a function of σ.

Convergence rates remained between 75% and 85% for noise levels up to 0.05 m, and more than two-thirds of runs still converged at $\sigma = 0.07$ m. Although variability across seeds increased with higher noise, overall learning stability was maintained.

These findings demonstrate that the policy tolerates substantial annotation noise without collapse and supports the feasibility of MR demonstrations under limited sensor accuracy or human precision.

F. Sim-to-Real Validation

To evaluate transferability from simulation to the physical system, trained policies were deployed on the cobot and tested with real objects in the workspace. Three evaluation conditions were considered: (1) simulation with ground-truth object poses, (2) real-robot execution with exact poses from the environment model, and (3) real-robot execution with object poses obtained through the HMD interface.

Figure 8 reports the success rates across these conditions. Both simulation and real execution with exact poses achieved

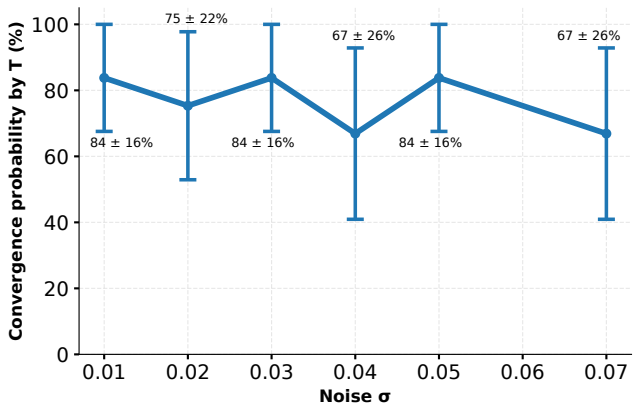

Fig. 7. Convergence probability under noisy expert annotations with Gaussian perturbations of varying standard deviation σ. The framework remained stable up to $\sigma = 0.05$ m, with reduced but still reliable convergence at higher noise levels.

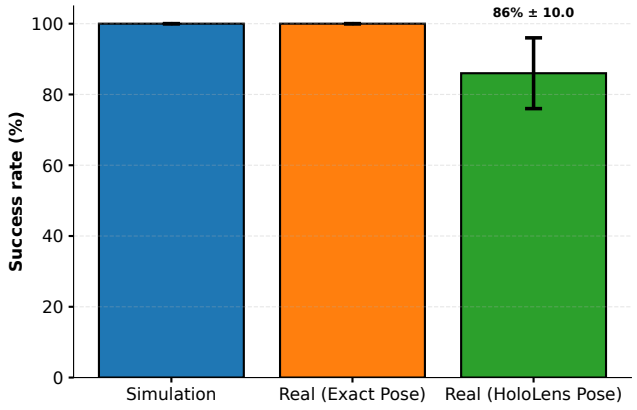

Fig. 8. Sim-to-real validation. Success rates for simulation with exact poses, real-robot execution with exact poses, and real-robot execution with HoloLens-derived poses. Minor degradation under MR input reflects robustness to sensor noise and calibration imperfections.

100% success, which confirms direct transfer when accurate state information is available. With HMD-provided poses, performance remained high at $86\% \pm 10\%$, despite sensor noise and calibration imperfections.

These findings indicate that workspace-aligned MR input enables policies to generalize reliably to the physical robot and effectively narrows the sim-to-real gap compared to conventional simulation-only training.

G. Discussion

The results demonstrate that MR demonstrations enhance learning across multiple dimensions, including learning speed, convergence reliability, generalization, robustness, and sim-to-real transfer. Taken together, these findings highlight the role of workspace-aligned expert guidance in stabilizing and accelerating policy optimization.

A central implication is the reduction of human effort during training. Figure 9 tracks human input during training. The red curve (left axis) shows the cumulative number of expert interventions. Because the pipeline accepts expert data only during the first half of training (10k steps, vertical dotted line), the red curve plateaus as intended. The green curve (right axis) shows the expert ratio per batch, defined as the fraction of samples drawn from the expert buffer in each minibatch. This ratio decreases progressively as training advances. Although the schedule was defined to decay linearly from 0.9 to 0.1, the realized ratio is lower in practice because the agent does not consistently require the maximum proportion of expert samples in the early phase. As shown in Figure 9, most training runs stabilized with an effective ratio of approximately 0.6, while only a few seeds relied on higher proportion of expert data at the beginning.

Shaded bands indicate variability across seeds. After being provided, expert actions are stored in a dedicated buffer and reused for matching observation states, which ensures that identical inputs do not need to be entered multiple times.

From a broader perspective, MR offers advantages over conventional demonstration methods. Unlike static datasets or VR teleoperation, MR annotations are spatially aligned with the physical workspace, that reduce ambiguity and support direct transfer to real robots. The robustness observed under noisy annotations further indicates that MR-based input remains effective despite sensor imprecision, which positions MR as a practical interface for human-in-the-loop DRL in structured tasks.

Certain limitations must be acknowledged. The evaluation was restricted to a single pick-and-place task and relied on manual annotations. Scaling to more complex tasks or higher-dimensional action spaces will require strategies to reduce annotation effort and to support more diverse forms of guidance. In addition, the reward function relied on environment-level variables, such as object and target positions. This design facilitated training but relied on privileged information that would need to be estimated through perception in broader deployment scenarios. The action space was defined in absolute Cartesian coordinates, which simplified the integration of MR annotations but differs from the incremental control typically used in RL for robotics. While this formulation was well suited to the present framework, extending it to relative actions could enhance applicability to tasks requiring fine-grained motion. These considerations highlight opportunities for future work to extend the framework toward more complex and less structured robotic applications.

VI. Conclusion

A demonstration-augmented RL framework was presented that integrates MR annotations into the training process of a cobot. Expert input was incorporated at the replay-buffer level through adaptive sampling and threshold-based activation, leading to faster convergence and reliable transfer from simulation to the physical robot. The results confirm that workspace-aligned MR guidance serves as an effective interface for human-in-the-loop DRL in structured robotic tasks. Future work may extend this approach to more complex manipulation problems and investigate richer forms of

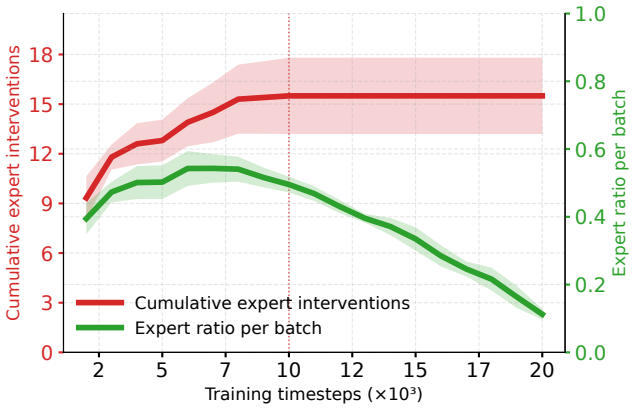

Fig. 9. Expert effort and sampling schedule. Red (left axis): cumulative expert interventions, plateauing after expert intake disabled at 10k steps (vertical dotted line). Green (right axis): expert ratio per batch, decreasing over time. Shaded bands: variability across seeds.

human feedback to support broader deployment of MR in robot learning.

ACKNOWLEDGMENT

The authors would like to thank the Federal Ministry of Education and Research (BMBF) for the financial support, as well as the Project Management Agency Karlsruhe (PTKA) for the administrative support of the collaborative project ”PerspektiveArbeit Lausitz (PAL)”.

REFERENCES

[1] F. Torabi, G. Warnell, and P. Stone, Behavioral cloning from observation, 2018, https://arxiv.org/abs/1805.01954.

[2] A. Correia and L. A. Alexandre, A survey of demonstration learning, Robotics and Autonomous Systems, 2024, https://doi.org/10.1016/j.robot.2024.104812.

[3] A. Rajeswaran, V. Kumar, A. Gupta, et al., Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2018, https://arxiv.org/abs/1709.10087.

[4] S. Gronauer and K. Diepold, Multi-agent deep reinforcement learning: a survey, Artificial Intelligence Review, 2022, https://doi.org/10.1007/s10462-021-09996-w.

[5] S. Levine, C. Finn, T. Darrell, and P. Abbeel, End-to-end training of deep visuomotor policies, arXiv, 2015, https://arxiv.org/abs/1504.00702.

[6] E. Yousefi, M. Chen, and I. Sharf, Baseline policy adapting and abstraction of shared autonomy for high-level robot operations, IEEE Transactions on Robotics, 2025, https://doi.org/10.1109/TRO.2025.3588455.

[7] J. Wu, Y. Zhou, H. Yang, Z. Huang, and C. Lv, Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, https://doi.org/10.1109/TPAMI.2023.3314762.

[8] Y. Zhang, C. Yan, J. Xiao, et al., NPE-DRL: enhancing perception constrained obstacle avoidance with nonexpert policy guided reinforcement learning, IEEE Transactions on Artificial Intelligence, 2025, https://doi.org/10.1109/TAI.2024.3464510.

[9] M. Vecerik, T. Hester, J. Scholz, et al., Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2018, https://arxiv.org/abs/1707.08817.

[10] Q. Wang, R. McCarthy, D. C. Bulens, et al., Improving behavioural cloning with positive unlabeled learning, 2023, https://arxiv.org/abs/2301.11734.

[11] S. Noh, S. Kim, and I. Jang, Efficient fine-tuning of behavior cloned policies with reinforcement learning from limited demonstrations, NeurIPS 2024, Workshop on Fine-Tuning in Machine Learning, 2024, https://openreview.net/forum?id=zKQ9RfuEuT.

[12] S. Sun, T. Li, X. Chen, et al., Cooperative defense of autonomous surface vessels with quantity disadvantage using behavior cloning and deep reinforcement learning, Applied Soft Computing, 2024, https://doi.org/10.1016/j.asoc.2024.111968.

[13] S. Kidera, K. Shintani, T. Tsuneda, and S. Yamane, Combined constraint on behavior cloning and discriminator in offline reinforcement learning, IEEE Access, vol. 12, pp. 19942-19951, 2024, https://doi.org/10.1109/ACCESS.2024.3361030.

[14] Z. Zhang, J. Hong, A. S. Enayati, and H. Najjaran, Using implicit behavior cloning and dynamic movement primitive to facilitate reinforcement learning for robot motion planning, 2024, https://arxiv.org/abs/2307.16062.

[15] P. Florence, C. Lynch, A. Zeng, et al., Implicit behavioral cloning, 2021, https://arxiv.org/abs/2109.00137.

[16] H. Zhu, A. Gupta, A. Rajeswaran, et al., Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost, 2018, https://arxiv.org/abs/1810.06045.

[17] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, Concept2Robot: Learning manipulation concepts from instructions and human demonstrations, The International Journal of Robotics Research, vol. 40, no. 12-14, pp. 1419-1434, 2021, https://doi.org/10.1177/02783649211046285.

[18] L. Guan, M. Verma, S. Guo, et al., Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation, 2021, https://arxiv.org/abs/2006.14804.

[19] L. Guan, M. Verma, S. Guo, et al., Explanation augmented feedback in human-in-the-loop reinforcement learning, in NeurIPS 2020 Workshop on Human and Model in the Loop Evaluation and Training Strategies, 2020, https://openreview.net/forum?id=20-xDadEYeU.

[20] S. Christen, S. Stevsic, and O. Hilliges, Demonstration-guided deep reinforcement learning of control policies for dexterous human-robot interaction, in 2019 International Conference on Robotics and Automation (ICRA), May 2019, pp. 2161–2167, https://doi.org/10.1109/ICRA.2019.8794065.

[21] A. Nair, A. Gupta, M. Dalal, and S. Levine, AWAC: Accelerating online reinforcement learning with offline datasets, 2021, https://arxiv.org/abs/2006.09359.

[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al., Continuous control with deep reinforcement learning, 2019, https://arxiv.org/abs/1509.02971.

[23] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, 2nd ed. Cambridge, MA: MIT Press, 2018.

[24] J. Kirkpatrick and et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the National Academy of Sciences, vol. 114, no. 13, pp. 3521–3526, 2017. [Online]. Available: http://dx.doi.org/10.1073/pnas.1611835114.

[25] S. Narvekar and et al., Curriculum learning for reinforcement learning domains: A framework and survey, 2020. [Online]. Available: https://arxiv.org/abs/2003.04960.

[26] C. Liu, Z. Zhang, D. Tang, et al., A mixed perception-based human-robot collaborative maintenance approach driven by augmented reality and online deep reinforcement learning, Robotics and Computer-Integrated Manufacturing, 2023, https://doi.org/10.1016/j.rcim.2023.102568.

[27] M. Mitra, G. Kumar, P. Chakrabarti, and P. Biswas, Investigating inverse reinforcement learning during rapid aiming movement in extended reality and human-robot interaction, J. Hum.-Robot Interact., 2025, https://doi.org/10.1145/3736423.

[28] V. V. R. Muvva, N. Adhikari, and A. D. Ghimire, Towards training an agent in augmented reality world with reinforcement learning, 17th International Conference on Control, Automation and Systems (ICCAS), 2017, https://doi.org/10.23919/ICCAS.2017.8204283.

[29] H. B. Mohammadi, M. A. Zamani, M. Kerzel, and S. Wermter, Mixed-reality deep reinforcement learning for a reach-to-grasp task, in Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation, pp. 611–623. [Online]. Available: https://doi.org/10.1007/978-3-030-30487-4_47

[30] C. Li, P. Zheng, Y. Yin, Y. M. Pang, and S. Huo, An AR-assisted deep reinforcement learning-based approach towards mutual-cognitive safe human-robot interaction, Robotics and Computer-Integrated Manufacturing, 2023, https://doi.org/10.1016/j.rcim.2022.102471.

[31] C. Li, P. Zheng, P. Zhou, Y. Yin, C. K. M. Lee, and L. Wang, Unleashing mixed-reality capability in deep reinforcement learning-based robot motion generation towards safe human–robot collaboration, Journal of Manufacturing Systems, 2024, https://doi.org/10.1016/j.jmsy.2024.03.015.