KhGRL: Kernelized human-Guided Reinforcement Learning

Edoardo Fiorini[1], Abhishek Padalkar[1], Antonin Raffin[1], and João Silvério[1]

***Abstract*— Learning from Demonstration (LfD) enables efficient synthesis of user-taught behaviors, while Reinforcement Learning (RL) allows autonomous skill acquisition in complex real-world environments. The Kernelized Guided Reinforcement Learning (KGRL) framework unifies these paradigms by guiding policy exploration using the covariance of user demonstrations and predefined hard constraints, thereby ensuring safety and sample-efficient learning. However, model uncertainty may in some cases lead to time-consuming exploration or to policies that are insufficiently generalizable when the environment changes. We extend KGRL by integrating human feedback that maps observation-space corrections into the corresponding action space and stores them in the replay buffer to accelerate exploration. Additionally, we propose a region-dependent action scaling factor learned via regression. This enables locally optimal exploration and compensates for reduced covariance guidance in low-uncertainty regions. We validate the method in a simulation environment that requires obstacle avoidance and goal-reaching.**

I. Introduction

Manipulation remains one of the most challenging problems in robotics. Various Learning from Demonstration (LfD) frameworks [1] generate robot motions that imitate and replicate user demonstrations. Movement Primitive (MP) methods—such as Kernelized Movement Primitives (KMPs) [2], Dynamic Movement Primitives (DMPs) [3], and Probabilistic Movement Primitives (ProMPs) [4]—generalize trajectories to address real-world challenges such as obstacle avoidance. However, these methods often struggle in dynamic or contact-rich environments, where demonstrations fail to fully capture task dynamics, resulting in out-of-distribution states and policy failures.

Reinforcement Learning (RL) [5] can address LfD limitations by training policies that account for both robot and environment states. However, most off-the-shelf RL solutions require a large number of trials, raising safety concerns and limiting direct deployment on real robots. Transfer learning can partially mitigate the issue by training in simulation, but it remains constrained by the sim-to-real gap [6]. Learning directly on real robots avoids this limitation, and guided RL approaches exploit available task knowledge—such as demonstrations and constraints—to improve policy exploration.

This work was supported in part by European Union's Horizon Research and Innovation Programme under Grant 101136067 (INVERSE) and in part by German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG). (*Corresponding author: Edoardo Fiorini.*)

[1]German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany.

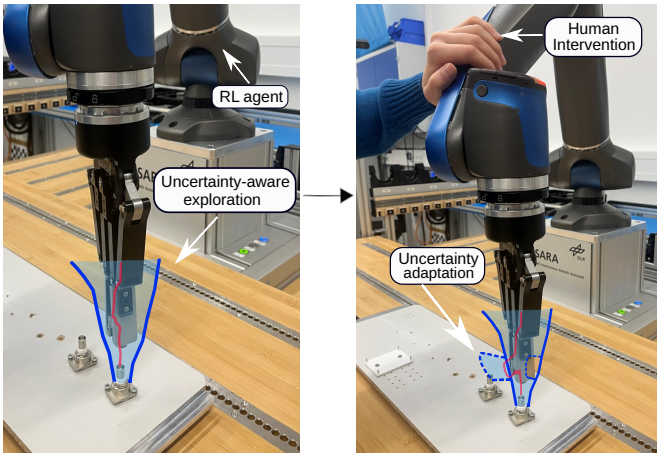

Fig. 1: Illustration of the proposed KhGRL framework on a robotic assembly task. The **RL agent** learns a policy for tasks with complex dynamics via **uncertainty-aware exploration** where both a nominal policy and state-dependent covariance are extracted from human demonstrations. In order to speed up and region-optimize the exploration our approach introduces online **human intervention** as a mechanism to locally and interactively modify the model uncertainty, changing the robot exploration behavior.

In [7], Linearly Constrained Null-Space Kernelized Movement Primitives (LC-NS-KMP), a non-parametric LfD method, are introduced. LC-NS-KMP generates motions that satisfy linear inequality constraints on the state of the robot. This formulation additionally leverages a *soft* null-space projector, introduced in [8], that allows RL actions to modulate the mean of the demonstrated policy based on the demonstrated variance (the lower/higher variance, the less/more modulation). Together with adhering to hard constraints, this property enables uncertainty-aware, state-dependent exploration while preserving safety. The resulting framework is called Kernelized Guided Reinforcement Learning (KGRL).

Though combining LfD guidance with RL promotes safe and adaptive policy exploration, human demonstrations may not always encode the best-suited exploration strategy for all parts of a task. For instance, in some parts, the user might be too consistent while demonstrating, despite the task allowing for variation, which limits exploration later if the task changes. Environmental changes may require additional data collection or retraining, and high model uncertainty can further slow the learning process.

Approaches for guiding reinforcement learning (RL) with human demonstrations have also been explored in prior work [9]–[11]. In these methods, demonstrations are incorporated into the replay buffer of off-policy algorithms to acceler-

ate learning. However, such approaches typically require reward signals to be recorded at every time step using the same reward function employed during training, ensuring consistency with the transitions stored in the replay buffer. If the reward function differs from the one used during training, the demonstrations become inconsistent and cannot be effectively utilized. In contrast, in KGRL, demonstrations are leveraged to learn a nominal policy and an uncertainty-aware, state-dependent exploration strategy. As a result, KGRL does not require reward information to be provided with the demonstrations.

In this paper, we propose an extension of the KGRL framework that incorporates human feedback to speed up policy learning and reduce the number of exploration steps. We retrieve corrective signals from the human intervention and map them to the corresponding RL action space. The corrected actions are then stored directly in the replay buffer. This mechanism, directly derived from the formulation of KGRL, improves exploration by biasing learning toward successful behaviors observed through human intervention, as highlighted in Fig. 1. Since the policy action space in KGRL differs from the robot state space (see Section II-B), incorporating human feedback requires inverting this relationship.

One additional aspect of KGRL lies in defining an exploration noise scaling factor that is appropriate for both low- and high-covariance regions. In the proposed framework, we address this challenge by learning the scaling factor from human feedback. For instance, in regions with high variance it may be difficult to generalize a skill if the environment changes after the demonstrations have been collected. In such cases, exploration can become excessively noisy, covering areas of the state space that are not necessary for the task. In summary, we:

1) We integrate human feedback into the KGRL framework by inverting the KGRL formulation (Section III-B) to map the agent's corrected behavior into the action space.
2) We leverage human intervention to estimate region-dependent exploration noise (Section III-C), improving policy learning efficiency.

We experimentally validate our approach in an illustrative 2D environment where an agent, simulating a mobile robot, must reach a goal by traversing two narrow passages while avoiding an obstacle. Results and key properties of the method are presented and discussed in Section IV. We conclude with potential future extensions in Section V.

II. Background

Let us define a set of M demonstrations $D = \{\{\boldsymbol{s}_{n,m}, \boldsymbol{\eta}_{n,m}\}_{n=1}^{N}\}_{m=1}^{M}$ where N is the length of a trajectory comprised of input $\boldsymbol{s} \in \mathbb{R}^{\mathcal{I}}$ and corresponding output $\boldsymbol{\eta} \in \mathbb{R}^{\mathcal{O}}$, where $\mathcal{I}$ and $\mathcal{O}$ denote the dimensions of the input and output spaces, respectively. This section introduces the methodological background underlying the proposed approach (Section III).

A. Probabilistic trajectory definition from demonstration

Gaussian Mixture Model (GMM) can be exploited to learn a probabilistic policy such that, $\mathcal{P}(s, \boldsymbol{\eta}) = \sum_{c=1}^{C} p_c \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where, p_c, $\boldsymbol{\mu}_c$, and $\boldsymbol{\Sigma}_c$ are the prior probability, mean and variance of the c^{th} Gaussian. To get a reference trajectory distribution $T_r = \{\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^{N}$ from the GMM, we can employ Gaussian Mixture Regression (GMR) to obtain a reference trajectory distribution where $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\Sigma}}_n$ are means and covariance matrices, respectively, computed at each new input $\hat{\boldsymbol{s}}_n$. At the same time, the set of demonstrations can be used to learn a parametric trajectory

$$\boldsymbol{\eta}(\boldsymbol{s}) = \boldsymbol{\Theta}(\boldsymbol{s})^{\top}\mathbf{w}, \quad \boldsymbol{\Theta}(\boldsymbol{s}) = \begin{bmatrix} \boldsymbol{\varphi}(\boldsymbol{s}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(\boldsymbol{s}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\varphi}(\boldsymbol{s}) \end{bmatrix}, \tag{1}$$

where the matrix $\boldsymbol{\Theta} \in \mathbb{R}^{\mathcal{B}\mathcal{O}\times\mathcal{O}}$, weight vector $\mathbf{w} \in \mathbb{R}^{\mathcal{B}\mathcal{O}}$, with $\boldsymbol{\varphi}(\boldsymbol{s})$ being a $\mathcal{B}$-dimensional basis function. Consider weights $\mathbf{w}$ are drawn from $\mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$, hence we can write $\boldsymbol{\eta}(\boldsymbol{s}) \sim \mathcal{N}(\boldsymbol{\Theta}(\boldsymbol{s})^{\top}\boldsymbol{\mu}_w, \boldsymbol{\Theta}(\boldsymbol{s})^{\top}\boldsymbol{\Sigma}_w\boldsymbol{\Theta}(\boldsymbol{s}))$.

B. Kernelized Guided Reinforcement Learning (KGRL)

KGRL [7] aims to derive a model of the demonstrations that captures the variance and correlations among data points while satisfying hard constraints. In addition, it introduces an uncertainty-aware term that supports exploration guided by the demonstration data. To achieve this, the method builds upon previous work on kernelized movement primitives [8], [12], [13] and formulates the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\mu}_w}{\textbf{argmin}} \quad & \sum_{n=1}^{N} \frac{1}{2}(\boldsymbol{\Theta}^{\top}(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n)^{\top}\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\Theta}^{\top}(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n) \\ & + \frac{1}{2}\lambda\boldsymbol{\mu}_w^{\top}\boldsymbol{\mu}_w + \frac{1}{2}\beta(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)^{\top}(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w), \\ \textbf{s.t.} \quad & \boldsymbol{g}_{n,f}^{\top}\boldsymbol{\eta}(\boldsymbol{s}_n) \geq c_{n,f}, \forall f \in \{1, 2, \dots, F\}, \\ & \forall n \in \{1, 2, \dots, N\}. \end{aligned} \tag{2}$$

where λ is a regularization factor, the term $\frac{1}{2}\lambda\boldsymbol{\mu}_w^{\top}\boldsymbol{\mu}_w$ regularizes the solution and the cost term $\frac{1}{2}\beta(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)^{\top}(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)$ inspired from [13] keeps the solution close to a desired one $\hat{\boldsymbol{\mu}}_w$. Additionally, the cost function is subject to linear inequality constraints, where F is the number of constraints imposed on the output and $\boldsymbol{g}_{n,f}$ and $c_{n,f}$ parameterize the constraint hyperplanes. Here, the main focus is to extract a policy for the robot to track. Equation (2) is solved by introducing Lagrange multiplier, replacing the solution in Eq. (1), and applying the *kernel trick* $\varphi(\boldsymbol{s}_i)^{\top}\varphi(\boldsymbol{s}_j) = k(\boldsymbol{s}_i, \boldsymbol{s}_j)$, we obtain [1]:

$$\begin{aligned} \tilde{L}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^{\top}\bar{\boldsymbol{G}}^{\top}\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{\mathcal{A}}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha} + (2\boldsymbol{\mu}^{\top}\boldsymbol{A}\boldsymbol{\mathcal{A}}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}} \\ - \beta\boldsymbol{\xi}^{\top}\underline{\boldsymbol{K}}^{-1}\hat{\boldsymbol{K}}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}} + \bar{\boldsymbol{C}}^{\top})\boldsymbol{\alpha} + \text{const}, \end{aligned} \tag{3}$$

[1] the reader is referred to [7] for a comprehensive derivation

and, for the expectation of (1),

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*)) = \boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha} + \frac{\beta}{\gamma}(\hat{\boldsymbol{k}}^* - \boldsymbol{k}^*\boldsymbol{A}\hat{\boldsymbol{K}})\underline{\boldsymbol{K}}^{-1}\boldsymbol{\xi}, \tag{4}$$

with $\boldsymbol{A} = (\boldsymbol{K} + \lambda\boldsymbol{\Sigma})^{-1}$, and $\mathcal{A} = -\frac{1}{2}\boldsymbol{K}\boldsymbol{\Sigma}^{-1}\boldsymbol{K} - \frac{\gamma}{2}\boldsymbol{K}$, where,

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{s}_1, \boldsymbol{s}_1) & \dots & \boldsymbol{k}(\boldsymbol{s}_1, \boldsymbol{s}_N) \\ \vdots & \ddots & \vdots \\ \boldsymbol{k}(\boldsymbol{s}_N, \boldsymbol{s}_1) & \dots & \boldsymbol{k}(\boldsymbol{s}_N, \boldsymbol{s}_N)) \end{bmatrix},$$

$$\boldsymbol{k}^* = [\boldsymbol{k}(\boldsymbol{s}^*, \boldsymbol{s}_1), \dots, \boldsymbol{k}(\boldsymbol{s}^*, \boldsymbol{s}_N)], \qquad \boldsymbol{k}(\boldsymbol{s}_i, \boldsymbol{s}_j) = k(\boldsymbol{s}_i, \boldsymbol{s}_j)\boldsymbol{I},$$

$$\underline{\boldsymbol{K}} = \hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\Phi}}, \quad \hat{\boldsymbol{K}} = \boldsymbol{\Phi}^\top\hat{\boldsymbol{\Phi}}, \quad \hat{\boldsymbol{k}} = \boldsymbol{\Phi}(\boldsymbol{s}^*)^\top\hat{\boldsymbol{\Phi}}, \quad \gamma = \lambda + \beta.$$

Moreover, $\boldsymbol{\xi} \in \mathbb{R}^{\mathcal{O}}$ is a term that can externally modulate the expectation of (1). As shown in [13], $\frac{\beta}{\gamma}(\hat{\boldsymbol{k}} - \boldsymbol{k}\boldsymbol{A}\hat{\boldsymbol{K}})\underline{\boldsymbol{K}}^{-1}$ acts as a *soft null space projector*, adapting the trajectory expectation in line with the data variance—with larger variance allowing stronger deformations and smaller variance constraining them. KGRL proposes to obtain *null space actions* $\hat{\boldsymbol{\xi}}$ from an RL policy $\pi(\hat{\boldsymbol{\xi}}|\boldsymbol{x})$, where $\boldsymbol{x}$ is the observed agent state, to modulate the LfD trajectory learned from the demonstrations[2]. Specifically, these actions modify the prediction for further refinement as:

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*)) = \boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha} + \frac{\beta}{\gamma}(\hat{\boldsymbol{k}}^* - \boldsymbol{k}^*\boldsymbol{A}\hat{\boldsymbol{K}})\pi(\hat{\boldsymbol{\xi}}|\boldsymbol{x}). \tag{5}$$

The kernel function $k(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is chosen according to the data. In this work, we employ the radial basis function (RBF) kernel [14].

III. KGRL WITH HUMAN-IN-THE-LOOP FRAMEWORK

In this section, we present the components of our Kernelized human-Guided Reinforcment Learning framework (KhGRL), which extends the Kernelized Guided Reinforcement Learning (KGRL) approach proposed in [7]. The framework integrates human intervention detection, null-space action estimation, and state-dependent exploration noise derived from user feedback. A detailed architecture diagram is shown in Fig. 2, which is primarily divided into two modules: the *Learner Process*, where the policy transitions are trained, and the *Actor Process*, where the agent interacts with the environment and human intervention is possible.

A. Human Intervention

During exploration, a human operator can inject domain knowledge at each time step by directly correcting the agent's actions in the continuous action space. We represent this human intervention in the observation space as $\boldsymbol{P}_h$, whose dimensionality matches that of the RL agent's action space. Such user feedback plays a crucial role in accelerating policy learning while maintaining safe exploration, as it guides the agent toward meaningful behaviors and prevents inefficient exploration of irrelevant regions of the state space. In this work, experiments are conducted in a custom 2D simulation environment, where human feedback is provided through direct interaction with the interface. Specifically, the user specifies corrections by clicking on the screen, and the system automatically infers the corresponding desired path deformations.

[2] $\boldsymbol{x}$ is used to represent the RL agent state, to distinguish it from the KMP input $\boldsymbol{s}$.

B. Null space action estimation from human feedback

In case of human intervention, given the detected $\boldsymbol{P}_h$, we derive the corresponding agent action $\hat{\boldsymbol{\xi}}$ at a specific state by inverting the KGRL expectation formulation in (5). Specifically, we assume the corrected expectation $\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*))$ to be equal to $\boldsymbol{P}_h$, as the output of KGRL directly represents the human-provided action correction. We assume that the human does not violate the constraints during the correction step and explicitly denote this case as $\boldsymbol{\alpha} = 0$ in (5). Under these assumptions, the resulting distribution is defined as follows:

$$\pi(\hat{\boldsymbol{\xi}}|\boldsymbol{x}) = (\frac{\beta}{\gamma}(\hat{\boldsymbol{k}}^* - \boldsymbol{k}^*\boldsymbol{A}\hat{\boldsymbol{K}}))^{-1}\boldsymbol{\Delta P} \tag{6}$$

where $\boldsymbol{\Delta P} = (\boldsymbol{P}_h - \boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\mu})$ captures the deviation of the human-provided correction from the KMP mean generated from the demonstrations.

The human-influenced computed $\hat{\boldsymbol{\xi}}$ is used to update the replay buffer in order to inject the right agent action.

C. State-dependent noise exploration from human feedback

In KGRL [7] every RL action generated by (5) is rescaled by a scaling factor in order to expand the exploration coverage region, injecting some noise. Specifically, the action $\hat{\boldsymbol{\xi}}$ is unscaled before being passed to KGRL to compute the final agent position $\boldsymbol{P}$. Conversely, in KGRL with human feedback, the observation-space $\boldsymbol{P}_h$, obtained from human intervention, is used to recover the corresponding action $\hat{\boldsymbol{\xi}}$, which must then be scaled accordingly, in order to ensure that the actions executed by the agent remain within a consistent and bounded range.

This design choice may introduce an implicit constraint when human intervention is present, as consistency within the replay buffer requires the exploration noise bounds to encompass the human-provided corrections. However, RL actions are modulated by both demonstration uncertainty and null-space projection. As a result, a human intervention occurring in a low-uncertainty region may lie outside the covariance of the demonstrations, leading to a large corrective action $\boldsymbol{P}_h$. In such cases, the corresponding exploration noise range may become excessively large, potentially destabilizing the learning process. Conversely, in regions of higher covariance, the same exploration noise magnitude may be either excessively large or overly restrictive, resulting in suboptimal exploration behavior.

Let us denote the exploration noise range as $\boldsymbol{L} = \{\ell_1, \dots, \ell_g\}, \quad g \in \{2, \dots, G\}$, where G has the same size of the agent action space $\hat{\boldsymbol{\xi}}$. We propose to learn its elements using a Gaussian Process Regression (GPR) [15] model as $\boldsymbol{L} = [\ell_1 = f(\hat{\xi}_1), \dots, \ell_g = f(\hat{\xi}_g)]$, which is initialized with default values and updated after each human intervention.

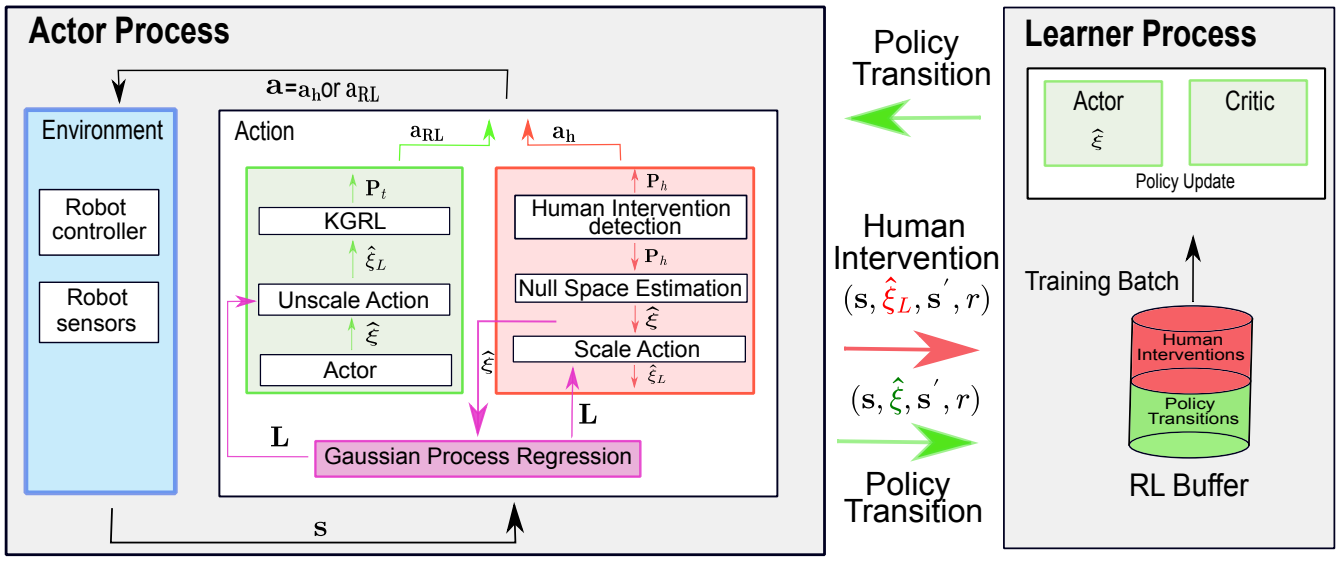

Fig. 2: Schematic overview of the proposed method. **Learner Process**: Optimize the policy using an actor-critic approach with a replay buffer composed of both human intervention transitions and autonomous policy-generated transitions. The updated policy is periodically deployed to the actor for exploration. **Actor Process**: Collect interaction data by executing the current policy in the environment, while allowing human intervention when necessary to guide exploration and prevent unsafe behaviors.

This formulation enables automatic, region-dependent refinement of the exploration limits based on human expertise, denoted by $\hat{\boldsymbol{\xi}}_L$. In this way, the user directly conveys to the agent the appropriate level of exploration noise required for effective policy learning. Indeed, as represented by the diagram in Fig. 2, this scaling range $\boldsymbol{L}$ is used to increase the magnitude of $\hat{\boldsymbol{\xi}}$ (*unscale*) in case of KGRL, while decreasing it (*scale*) in case of KhGRL.

IV. Evaluation and Discussion

We evaluate our approach in a custom 2D simulation environment through two scenario experiments. In the first scenario, a narrow passage and circle obstacle is located within a region of high demonstration covariance, as highlighted in light grey and brown, respectively, in **??**-a. In the second scenario, the same narrow passage, illustrated in **??**-b, is placed in a low-covariance region.

As shown by dashed blue lines in **??**, the same set of demonstrations $D = \{\{\boldsymbol{s}_{n,m}, \boldsymbol{\eta}_{n,m}\}_{n=1}^{400}\}_{m=1}^{9}$ is used in both experiments, while the red line illustrates the retrieved mean trajectory exploited in (5). We compare our approach against KGRL as the baseline.

In our method, the 2D robot's desired position is computed as $\boldsymbol{P}_t = \mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*))$, which can follow either the time-based trajectory described in Section II-B, or a human intervention defined as $\boldsymbol{P}_t = P_h$. In the latter case, the corresponding action is retrieved, and the buffer is updated accordingly. Conversely, the baseline *vanilla* KGRL follows the policy defined by (5).

The human can provide feedback interactively by dragging the desired correction of the agent position directly within the simulation environment, as illustrated by the black line in **??**. The corrected position is automatically retrieved and assigned to $\boldsymbol{P}_t$. In both cases, the human provides corrections for seven specific episodes: $4, 5, 10, 19, 20, 28$, and 29.

During the training phase, an episode is considered successful if the robot reaches the goal within 400 time steps. Otherwise, the episode terminates unsuccessfully if the robot remains blocked in one of the narrow passages for at least 20 consecutive time steps, or if the 400-step limit is reached without reaching the goal.

The RL policy is learned using a neural network with 2 hidden layers with 256 neurons each, in both baselines. To train the RL policies, we used the implementation of Truncated Quantile Critics (TQC) [16] from Stable-Baselines3 [17], with parameters as follows: *learning rate = 0.001, soft update coefficient = 0.02, discount factor = 0.99, training frequency = 8, gradient steps = 8, entropy regularization coefficient = automatically adjusted*.

A. Narrow Passage in a High-Covariance Region

This experiment aims to demonstrate how human feedback can accelerate the RL policy learning through interactive interventions. The aim of our framework is to exploit users' prior knowledge to enable the agent to more effectively avoid both the obstacle and the narrow passage, which is challenging for pure KGRL. This difficulty arises because the exploration noise, which is kept equal across both approaches, leads to significant deviations in regions with high demonstration covariance. Consequently, the agent frequently collides with the obstacle. The performance comparison between KhGRL and vanilla KGRL is reported in Fig. 3. The exploration noise range is set as $\boldsymbol{L} = \{20, 20\}$.

In this case, the robot's reward function is defined as

$$r_t = r_a + r_o + r_p + r_T, \qquad r_a = -10\delta \boldsymbol{p}_t^\top \delta \boldsymbol{p}_t, \tag{7}$$

$$r_o = \begin{cases} -10, & \text{if } \boldsymbol{d}_t \text{is inside the obstacle} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$r_p = \begin{cases} -10, & \text{if } \boldsymbol{d}_t \text{is inside the passage} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$r_T = \begin{cases} 200 & \text{at terminal step } T \text{ if successful} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $\boldsymbol{d}_t$ denotes the distance between the robot and the obstacle, r_T is the terminal reward provided upon successful episode completion, r_o represents the obstacle avoidance penalty, r_p is the passage avoidance penalty and r_a denotes the action cost.

B. Narrow Passage in a Low-Covariance Region

In this scenario, we want to highlight the effectiveness of human feedback in defining a region-dependent optimal $\boldsymbol{L}$. The human can intervene in the narrow grey passage, which is now located in a region of low covariance. The human feedback is collected and modeled through a GPR that tracks the $\hat{\boldsymbol{\xi}}$ retrieved from his input, as described in Section III-B. This mechanism is crucial for successfully completing the task and reaching the goal. In contrast, in the *vanilla* KGRL case, the agent is unable to explore outside the covariance region and therefore fails to avoid the obstacle. Fig. 4 shows the performance comparison between the two considered approaches in generating a policy in a low-covariance region. We set an initial $\boldsymbol{L} = \{15, 15\}$, and iterate it after the first human correction has been given. This time, since there is only a narrow passage, the robot's reward function is defined as

$$r_t = r_a + r_p + r_T \tag{11}$$

C. Discussion

The center plot of Fig. 3 compares obstacle costs in the first simulation environment, with blue and red curves representing KGRL with and without human intervention, respectively. KhGRL converges to zero cost significantly faster, enabling the robot to learn a successful policy in fewer episodes. This improvement arises from directly injecting successful RL action trajectories into the replay buffer via human guidance (Eq. (6)). Covariance analysis further highlights this effect: vanilla KGRL shows higher variability due to extensive exploration, whereas KhGRL converges more stably. The top plot of Fig. 3 confirms this trend in rewards. However, KGRL experiences more collisions during learning, as represented by high obstacle cost. Overall, these results demonstrate that human-guided experience significantly accelerates policy learning.

Figure 4 shows the cost function behavior in the second simulation environment, where KGRL with human intervention (blue line) reaches convergence in fewer episodes, whereas pure KGRL (red line) fails to learn a viable policy. Learning the optimal exploration noise range $\boldsymbol{L}$ from human feedback using GPR, our approach defines a region-dependent exploration noise that overcomes the low-covariance limitation, enabling safe exploration in other regions of the task—something not possible with a default large action noise. Figure 5 compares the evolution of the estimated scaling factors for KhGRL and vanilla KGRL during a representative exploration episode. In the KhGRL case, the scaling factor exhibits a clear deviation from its default value within the episode interval 100–300, which corresponds to the region of the task where user intervention is applied. This behavior indicates that KhGRL adapts its scaling in response to external guidance in low-covariance regions. In contrast, *vanilla* KGRL maintains the default constant set scaling factor throughout the entire episode. This highlights the increased responsiveness and adaptability of KhGRL compared to the baseline approach.

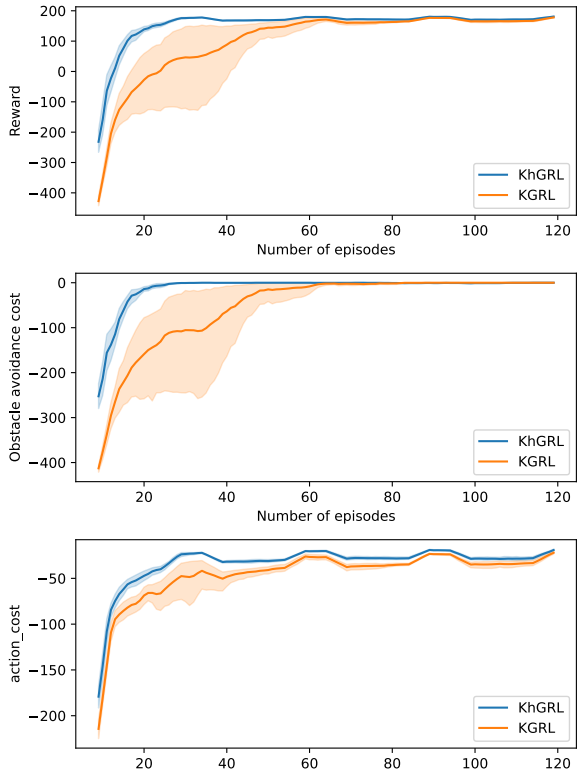

Fig. 3: Comparison of the performance of KGRL and KhGRL, when the correction is given in a high-covariance region.

V. Conclusions

This paper presents an extension of the KGRL framework that incorporates human feedback directly into the replay buffer, enabling it to influence the reinforcement learning actor–critic mechanism during training. The proposed method exploits human interventions expressed in the agent state space to retrieve the corresponding agent action, derived from the KGRL formulation. By mapping this information into the RL action space, the agent can leverage successful human-guided trials to accelerate learning.

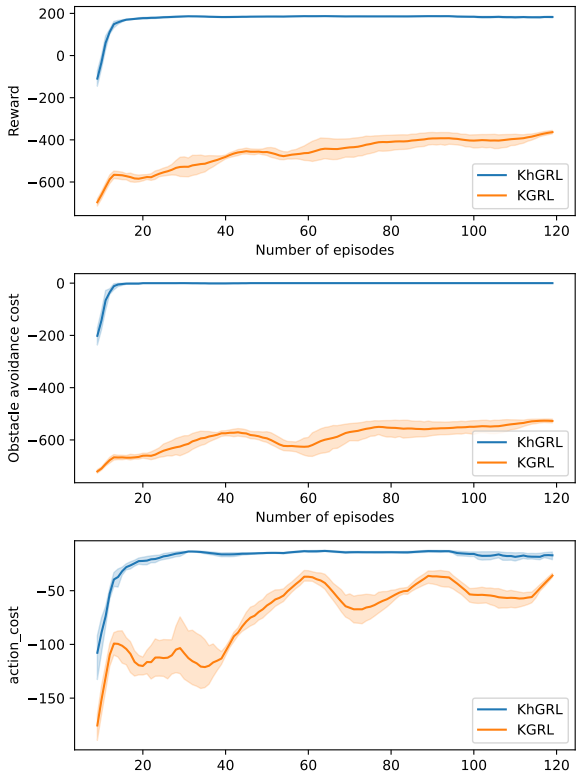

Fig. 4: Comparison of the performance of KGRL and KhGRL, when the intervention is applied in a low-covariance region.

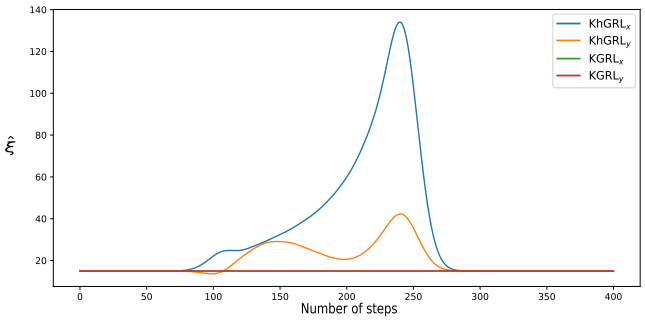

Fig. 5: Comparison of the scaling factor elements $\hat{\xi}$ in the set $\boldsymbol{L}$ for KhGRL and KGRL, when the narrow passage is in low-covariance region.

Additionally, we introduced an online exploration tuning strategy that adapts the RL exploration noise through a region-based scaling factor estimated in real time from human feedback. This estimation is performed using Gaussian Process Regression (GPR), enabling the agent to dynamically adjust its exploration behavior.

The framework was evaluated in a custom 2D simulation environment where a robot navigates toward a target pose while avoiding obstacles. Human feedback is provided interactively through direct user input within the simulator.

Future work will focus on integrating mechanisms for detecting physical human–robot interaction and extending the proposed framework to real-world experiments using collaborative robots.

References

[1] H. Ravichandar, A. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 05 2020.

[2] Y. Huang, L. Rozo, J. a. Silvério, and D. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research (IJRR*, vol. 38, pp. 833–852, 05 2019.

[3] S. Schaal, *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*. Tokyo: Springer Tokyo, 2006, p. 261–280.

[4] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[5] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2024.

[6] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning," *IEEE Access*, vol. PP, pp. 1–1, 11 2021.

[7] A. Padalkar, F. Stulp, G. Neumann, and J. a. Silvério, "Towards safe and efficient learning in the wild: Guiding rl with constrained uncertainty-aware movement primitives," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6880–6887, 2025.

[8] Y. Huang and D. G. Caldwell, "A linearly constrained nonparametric framework for imitation learning," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4400–4406.

[9] J. Luo, O. Sushkov, R. Pevceviciute, W. Lian, C. Su, M. Vecerik, N. Ye, S. Schaal, and J. Scholz, "Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study," *arXiv preprint arXiv:2103.11512*, 2021.

[10] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.

[11] J. Stranghöner, P. Hartmann, M. Braun, S. Wrede, and K. Neumann, "Share-rl: Structured, interactive reinforcement learning for contact-rich industrial assembly tasks," 2025. [Online]. Available: https://arxiv.org/abs/2509.13949

[12] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *International Journal of Robotics Research (IJRR)*, vol. 38, no. 7, pp. 833–852, 2019.

[13] J. Silvério and Y. Huang, "A non-parametric skill representation with soft null space projectors for fast generalization," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2988–2994.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[15] ——, *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.

[16] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5556–5566.

[17] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.