Repairing Neural Networks for Safety in Robotic Systems using Predictive Models

Keyvan Majd[1], Geoffrey Clark[1], Georgios Fainekos[2], and Heni Ben Amor[1]

***Abstract*— This paper introduces a new method for safety-aware robot learning, focusing on repairing policies using predictive models. Our method combines behavioral cloning with neural network repair in a two-step supervised learning framework. It first learns a policy from expert demonstrations and then applies repair subject to predictive models to enforce safety constraints. The predictive models can encompass various aspects relevant to robot learning applications, such as proprioceptive states and collision likelihood. Our experimental results demonstrate that the learned policy successfully adheres to a predefined set of safety constraints on two applications: mobile robot navigation, and real-world lower-leg prostheses. Additionally, we have shown that our method effectively reduces repeated interaction with the robot, leading to substantial time savings during the learning process.**

I. Introduction

Robot learning holds great promise for advancing wearable robotics, such as Prosthetics, Orthoses, and Exoskeletons [1]. These devices are typically designed with a one-size-fits-all approach. Deep learning techniques offer the potential to customize policies for each user that leads to improved comfort and quality of life. One common approach for customizing policies has been through reinforcement learning (RL). However, RL can be time-consuming and potentially hazardous in real-world scenarios, as agents must explore the environment and learn from mistakes [2]. In our previous work [3], we introduced an algorithm for training neural network policies that satisfy given safety specifications using neural network repair. By solving a layer-wise Mixed-integer Quadratic Program (MIQP), we modified network weights to meet desired safety constraints on outputs while providing mathematical guarantees on safety for the samples used in repair. However, our method primarily focused on addressing explicit safety constraints over the output space of the neural network. Building upon our previous work, we now extend our method to address implicit safety constraints on the output of predictive models.

In this paper, we present **SARP** (**S**afety-**A**ware **R**epair with **P**redictive models), an approach for safety-driven learning of robot policies from human demonstrations. Our method focuses on repairing an existing neural network policy to ensure compliance with a defined set of safety constraints. We leverage a predictive model to predict the features of the system given states and actions. These features can encompass various aspects in different robot learning applications, such as proprioceptive states in biomechanical applications, or collision occurrence in navigation scenarios.

In a two-step supervised learning process, SARP initially trains a policy using expert demonstrations. We then employ a predictive model to enforce safety constraints on the policy. This is achieved through the application of neural network repair techniques to regulate the predicted system features. SARP addresses both implicit constraints on the policy, using differentiable predictive models, and explicit bounding constraints on actions. We assess our method in two safety-critical case studies. We first compare SARP's performance with two state-of-the-art safe RL methods, as presented in [4] and [5]. This comparison aims to demonstrate SARP's efficacy and comparable performance in a simulated mobile robot navigation task as a showcase example. Our goal is not to position SARP as a substitute for RL but rather as a complementary method that addresses the safety concerns associated with lengthy simulation in RL for safety-critical robot learning tasks, such as lower-leg prosthesis control. We finally test SARP for controlling a real-world lower-leg prosthesis in an IRB-approved study.

Our novel **contribution** lies in repairing the policy to ensure that the predicted features of the system adhere to a predefined set of safety constraints. Specifically, this paper makes the following contributions:

1) We propose a novel two-step supervised learning method that combines imitation learning and neural network repair using predictive models.
2) Our framework can utilize any pre-existing differentiable model in the repair process that reduces the repetitive interactions with the robot.
3) We compare SARP with two safe RL-based methods of constrained policy optimization (CPO) in [4] and the safety layer approach in [5].
4) The code for reproducing the results is available at `https://github.com/k1majd/SARP.git`.

II. Related Work

A. Safe Reinforcement Learning

Safety in robot learning has been extensively explored in the existing literature [2]. One category of methods focuses on learning uncertain dynamics certifying safety. These methods typically rely on prior knowledge of the system dynamics [6]–[8] or learn the uncertainties in the dynamics [9]–[11]. Another category encourages safety during policy exploration or penalizes dangerous actions in a model-free fashion. Safety exploration strategies include learning a

[1] K. Majd, G. Clark, and H. Ben Amor {majd, gmclark1, hbenamor}@asu.edu are with SCAI, Arizona State University, Tempe, AZ, USA.

[2] G. Fainekos is with Toyota NA R&D, Ann Arbor, MI, USA.

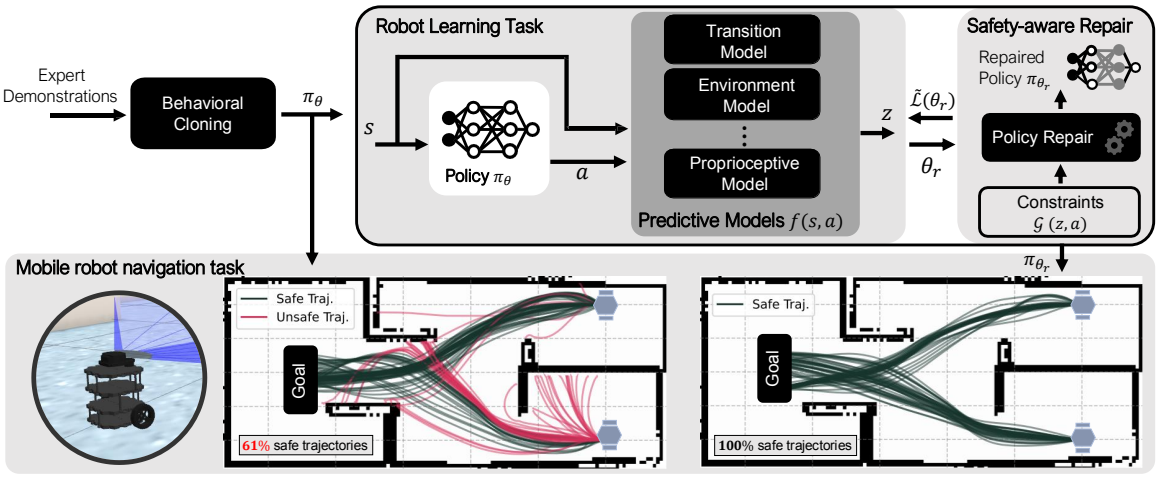

Fig. 1: Safety-Aware Repair with Predictive models (SARP). Left: A policy trained for a mobile navigation task (point to goal motion planning in this figure). Right: Policy repair module that adjusts the policy parameters to penalize unsafe behavior, based on a set of safety constraints and the loss of predictive models. Predictive models may include state-action transition model, a model of environment, or a proprioceptive model predicting internal states of the system.

safety critic function [12], taking risk-averse action through learning an ensemble model [13], [14], or editing the policy through a safety layer [5], [15]. A closely related line of work to ours involves employing RL approaches to solve a relaxed Constrained Markov Decision Process (CMDP) using Lagrangian relaxation [4], [16], [17]. These techniques use primal-dual updates on reward and constraint surrogates to ensure policy improvement and near-constraint satisfaction.

In RL, finding an effective reward-shaping scheme that promotes safe behavior while achieving desired performance objectives is a non-trivial task. Actions that lead to high rewards may also increase the risk of safety violations. Moreover, RL methods require extensive simulation or robot-environment interaction time to converge and to learn optimal policies. Such time-consuming learning process is especially problematic in safety-critical scenarios involving physical human-robot interaction, such as with robotic prostheses devices. Repeated interactions between humans and robots to reach a safe solution can pose a risk to the human user. In contrast, SARP combines Behavioral Cloning (BC) with neural network repair in a two-step supervised learning framework. Our method reduces the necessity for lengthy simulations, allowing for easier offline repair of policy for safety. SARP adds to the tools of robotics engineers when RL approaches are deemed too expensive or potentially unsafe, such as in physical human-robot interactions.

B. Neural Network Repair

Another relevant area of research related to this paper is neural network (NN) repair. One approach to repair is repeated retraining and fine-tuning based on counterexamples [18]–[20]. However, this approach requires the availability of counterexample labels, which may involve additional data collection. Several approaches exist for provable repair of networks that can guarantee safety for either faulty samples or faulty input-output linear regions of the network [3], [20]–[25]. These methods are only applicable to networks with piecewise linear activation functions and only address explicit constraints over the network's output.

Our repair method extends beyond the scope of piecewise-linear networks and offers repair to both explicit and implicit constraints on the output of the network. We propose employing a differentiable predictive model in repairing a pre-trained NN policy. It enables the incorporation of task-specific features such as collision prediction or proprioceptive features as implicit constraints on the policy.

III. Problem Formulation

While learning the policy directly from expert demonstrations may not guarantee a safe policy, safety can be attained by specifying and enforcing safety constraints that the policy must adhere to. Our method guides the policy toward safety through predictive models that predict the future system features. A predictive model can have different forms. It might be derived analytically or learned from expert demonstrations and robot exploration samples. A predictive model can learn static environmental properties such as friction and surface characteristics, or it may be a proprioceptive model that estimates the internal states based on sensor outputs.

Our approach in this work proposes a safety-guided policy repair method that employs a set of safety constraints on either the predictive features of system or actions of the policy. Figure 1 shows an example scenario in which a robot learns from demonstrations to navigate from multiple start locations to a goal. First, we pre-train a NN policy π_θ with parameters $\theta \in \Theta$ using imitation learning to learn from expert demonstrations, where Θ represents the network's parameter space. As shown in Fig. 1, the pre-trained policy π_θ generates about 39% unsafe trajectories, i.e., robot actions lead to collisions. Next, we employ a differentiable predictive model that predicts the occurrence of collision in future time steps to guide the pre-trained policy

towards safety. This is achieved by optimizing the policy parameters θ through a NN repair module that penalizes unsafe behavior, as shown in Fig. 1 (right). This safety-guided repair step is performed in an offline optimization and does not involve iterative interaction with the environment. The resulting repaired parameters θ_r lead to a policy that satisfies the safety constraints over the training data.

Problem 1 (Safety-Aware Repair with Predictive models)**.** *Let $f\colon \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ be a differentiable predictive model that predicts a feature $z \in \mathcal{Z} \subseteq \mathbb{R}^{n_\mathcal{Z}}$ of the system given the previous state $s \in \mathcal{S} \subseteq \mathbb{R}^{n_\mathcal{S}}$ and action $a \in \mathcal{A} \subseteq \mathbb{R}^{n_\mathcal{A}}$. Here, $\mathcal{Z}$, $\mathcal{S}$, and $\mathcal{A}$ denote the feature, state, and action spaces, respectively, with dimensions $n_\mathcal{Z}$, $n_\mathcal{S}$, and $n_\mathcal{A}$. Consider a trained NN policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$, and a differentiable predictive model $f(s,a)$, which predicts a system feature z. Let $\mathcal{G}(z,a)$ denote a set of constraints that must hold true for the robot learning task. The goal of safety-aware repair is to adjust the policy parameters θ so as to minimize the loss function $\mathcal{L} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ while satisfying $\mathcal{G}(z,a)$.*

IV. Safety-guided Policy Repair using Predictive Models

Our method involves formulating and solving an optimization problem that minimizes a loss function subject to safety constraints over the predicted system features and actions. The features of the system could represent a non-observable or future state of the system, or other system parameters, such as friction, mass, etc. SARP aims to modify the parameters θ of the policy π_θ to adjust the feature $z = f(s,a)$ and the actions $a = \pi_\theta(s)$ such that the set of constraints $\mathcal{G}(f(s,a),a)$ is satisfied for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Due to the non-convex nature of the optimization problem and the nonlinear activation functions of the policy, finding a global solution can be challenging. To address this issue, the works in [3] and [25] proposed a layer-wise relaxation approach that focuses on repairing the networks with Rectified Linear Unit (ReLU) activation functions. In particular, assuming quadratic loss functions and modifying the weights of a single layer, this method formulates repair as a mixed-integer quadratic program (MIQP) that can be solved using off-the-shelf solvers [3], [25]. This method also assumes explicit constraints on the output of policy network.

This paper presents an extended approach to repairing the policy network π_θ, inspired by the previous methodologies [3] and [25]. In addition to addressing explicit constraints on the output of the policy network, our method includes constraints associated with the system's features z, as defined by the predictive model $f(s,a)$. Unlike [3] and [25], which focused on a single layer repair and were limited to only (Piecewise-)Linear activation functions, our method offers a more comprehensive solution. We do not restrict ourselves to a particular class of activation functions and we enable repair across multiple layers of the policy network. We can

Algorithm 1 Safety-aware repair with predictive models

Input: $\pi_\theta, f, \tilde{\mathcal{L}}, \mathcal{G}, \mathcal{D}$
Output: π_{θ_r}
1: Initialize $\mu_0 \in \mathbb{R}_+, \lambda_0 \leftarrow 0, k \leftarrow 0, \theta_r \leftarrow \theta$
2: **while** SafetyCheck$(\theta_r, f, \mathcal{G}, \mathcal{D}) \neq$ **true do**
3: $\quad\theta_r \leftarrow \arg\min_{\theta_r} \tilde{\mathcal{L}}(\pi_\theta(s), \lambda_k, \mu_k)$, for $(s,a) \in \mathcal{D}$
4: $\quad\lambda_{k+1} \leftarrow -ReLU\Big(\eta \sum_{c=1}^{C} g_c(f(s, \pi_{\theta_r}(s))) - \lambda_k\Big)$
5: $\quad\mu_{k+1} \leftarrow \beta \mu_k$
6: $\quad k \leftarrow k+1$
7: **end while**

formulate Problem 1 as an optimization problem:

$$\theta_r = \arg\min_{\theta \in \Theta} \ \mathcal{L}(\pi_\theta) \tag{1}$$

$$\text{s.t. } z = f(s, \pi_\theta) \ \ \forall s \in \mathcal{S}, \tag{2}$$

$$\mathcal{G}(z, \pi_\theta). \tag{3}$$

This problem optimizes θ by minimizing the loss function (1), while satisfying the constraints (2) and (3). In this paper, we assume that the set of constraints (3) is defined as a conjunction of equality constraints of form $\bigwedge_{c=1}^{C} g_c(z, \pi_\theta) = 0$, where C represents the number of constraints and $g_c : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$ is a differentiable nonlinear function dependent on variables z and a. Since constraints in practical scenarios often involve inequalities, we handle inequality constraints of the form $g(\cdot) \leq 0$, we employ a replacement approach by transforming them into a rectified linear unit (ReLU) function: $ReLU(g(\cdot)) = \max\{0, g(\cdot)\}$. This modification effectively allows us to address the inequality constraints, as $ReLU(g(\cdot)) = 0 \iff g(\cdot) \leq 0$. Given that $z = f(s, \pi_\theta(s))$, we can integrate the constraints (2) and (3) by $\bigwedge_{c=1}^{C} g_c(f(s, \pi_\theta(s)), \pi_\theta(s)) = 0$. We further relax the constraints by incorporating them into the loss function (1) inspired by the Augmented Lagrangian Method (ALM) [26]. The modified loss with incorporated constraints is given by:

$$\tilde{\mathcal{L}}(\theta, \lambda, \mu) = \mathcal{L}(\pi_\theta) - \sum_{c=1}^{C} \lambda_c g_c\Big(f(s, \pi_\theta), \pi_\theta\Big) + \frac{\mu}{2} \sum_{c=1}^{C} g_c^2\Big(f(s, \pi_\theta), \pi_\theta\Big). \tag{4}$$

Here, $\sum_{c=1}^{C} g_c(\cdot)$ represents the penalty terms with Lagrange multipliers $\lambda \in \mathbb{R}^C$, that balance the satisfaction of the constraints and the overall objective optimization. The parameter $\mu \in \mathbb{R}$ scales the quadratic penalty terms $\sum_{c=1}^{C} g_c^2(\cdot)$, that penalize the violations of constraints and encourage the optimization process to satisfy them. We formulate the safety-aware policy repair as follows:

SARP Optimization. *Let π_θ denote a trained policy, and let z be a feature of the system given by $f(s,a)$ as defined in (2). Let also $\bigwedge_{c=1}^{C} g_c(z,a) = 0$ be the constraints on the feature z for the policy action $a = \pi_\theta(s)$, where C represents the number of constraints, $g_c(\cdot)$ is a differentiable nonlinear*

Fig. 2: A simulation of a mobile robot in a hospital scenario. The robot is tasked with getting to different rooms without colliding with the environment.

function, and $s \in \mathcal{S}$. Safety-aware policy repair optimizes θ such that the augmented loss function (4) is minimized:

$$\theta_r = \underset{\theta \in \Theta, \mu \in \mathbb{R}, \lambda \in \mathbb{R}^C}{\arg\min} \tilde{\mathcal{L}}(\theta, \lambda, \mu). \tag{5}$$

To solve the problem (5), we present Alg. 1 that is inspired by the iterative primal-dual algorithm [26]. Consider a collection of N samples denoted as $\mathcal{D} = \{(s_n, a_n)\}_{n=1}^{N}$, that represent the expert's demonstrations. Here, $(s_n, a_n) \in \mathcal{S} \times \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ denote the sets of states and actions, respectively. We further define π_θ be a trained policy over the expert's samples $\mathcal{D}$. In Alg. 1, we initialize λ_0 as 0 and μ_0 as a positive real number ($\mu \in \mathbb{R}_+$). At each iteration k, the algorithm modifies the parameters of π_θ to minimize the augmented loss (4) for all state-action tuples in $\mathcal{D}$. We employ Stochastic Gradient Descent [27] as our chosen optimization algorithm to solve (5). Next, the algorithm updates the Lagrange multiplier λ_k by dual ascent as in line 4. Additionally, the penalty coefficient μ_k is updated according to line 5 of the algorithm. Here, $\eta \in \mathbb{R}_+$ and $\beta \in [1, \infty)$. To assess the safety of the repaired policy π_{θ_r} on the training set $\mathcal{D}$, Alg. 1 employs the function SAFETYCHECK$(\cdot)$ at each iteration. This function terminates repair if the safety condition is satisfied for all samples. Note that the primary factor affecting computational cost is the number of optimization variables θ. The convergence rate can also be influenced by the number of violated constraints, but the impact is problem-dependent.

Remark 1. *In addition to verifying safety for all samples in $\mathcal{D}$ using* SAFETYCHECK$(\cdot)$*, we can further validate that the property $\mathcal{G}$ holds over a broader operating domain $S' \times A' \subseteq S \times A$ using a neural network reachability tool [23]. This approach not only confirms constraint satisfaction for regions beyond the collected sample set but also allows us to add any violating samples to $\mathcal{D}$ for further robustness.*

V. APPLICATION SCENARIOS

In this section, we evaluate SARP by presenting its application in two safety-critical scenarios. The first example is presented as a simulation showcase, comparing SARP with two state-of-the-art safe RL approaches: Constrained Policy Optimization (CPO) [4] and the safety-layer approach (DDPG+SL) [5]. In this scenario, we demonstrate the application of SARP to a mobile robot navigation task in a hospital simulation, see Fig. 2. Our repair method guides the robot towards safe behavior by incorporating safety constraints on its predicted actions and the outputs of a trained predictive model that predicts possible collisions using range sensor readings. By doing so, we mitigate the risk of accidents or collisions, which is crucial for robot navigation in a safety-critical environment such as a hospital.

Fig. 3: Lower-leg prosthesis system. (left) Image depicts the upper limb angle α_{ul}, the lower limb angle α_{ll}, and the ankle angle α_a. (middle) The location of pressure sensors p_1-p_{16}. (right) The robotic lower-limb prosthesis device.

The second scenario demonstrates a real-world application of SARP in controlling a lower-leg prosthesis device. Recently, machine learning has been shown to be a promising approach in enhancing the control and functionality of lower-leg prosthetic devices [28]. Here we aim to limit the action rate and the pressure applied to the foot, see Fig. 3 (right). Measuring foot forces in real-time can be expensive due to several factors, such as the need for multiple sensors for accurate measurements as well as the complexity of integration with the prosthesis and customized software. Hence, we utilize a predictive model to estimate the pressures applied to the foot. We demonstrate that SARP promotes the safe operation of the prosthesis device by guiding it to behave within safe boundaries through enforcing safety constraints on the predicted actions and bio-mechanical states of the system. In all experiments, we defined $\mathcal{L} = ||\pi_\theta - \pi_{\theta_{IL}}||^2$. Throughout our experiments, we calculate all relative improvement metrics, including Relative Change (RC) and Side Effect (SE), using the following formula (assuming the target value is x): $(x_{new} - x_{orig})/(x_{orig}) \times 100\%$.

A. Showcase Example: Robot Navigation in Hospital

In this task, a NN policy is trained to control the linear and angular velocities of a robot, denoted as $a = [v(t), \omega(t)]^T$. The system states include the robot's Cartesian goal location $[x_g, y_g]^T$, Euclidean distance to the goal d_g, heading towards the goal ϕ_g, and a 2D range sensor readings vector $r = [r_1, \cdots, r_M]^T$, with M representing the number of sensor rays ($M = 10$ in our experiments). Hence, the system state is defined as $s = [x_g(t), y_g(t), d_g(t), \phi_g(t), r(t)]^T$. To predict collisions, we utilize a predictive model $f(r(t))$ that takes the range sensor readings at time t and predicts a collision occurrence at $t + 1$, i.e., $z = \{0, 1\}$, where 1 indicates a collision and 0 indicates no collision. In this experiment, we

TABLE I: Robot navigation statistics. GR: average percentage of the trajectories that reached the goal, E: average percentage of safe samples (Efficacy), and ST: simulation time. The results are the average of 200 test trajectories.

Method	GR (Collision)	E (Collision)	ST [h]
Orig. Policy	68.7%	74.4%	$0_{+0.5}$
$\text{SARP}_{\text{penalty}}$	89.8%	92.7%	$\mathbf{3}_{+0.5}$
$\text{SARP}_{\text{Lagrangian}}$	**98.1%**	97.8%	$\mathbf{3}_{+0.5}$
$\text{CPO}_{\text{sparse, 3h}}$	74.4%	99.3%	$\mathbf{3}_{+0.5}$
$\text{CPO}_{\text{dense, 3h}}$	75.0%	97.2%	$\mathbf{3}_{+0.5}$
$\text{DDPG+SL}_{\text{sparse, 3h}}$	78.2%	99.8%	$\mathbf{3}+3_{+0.5}$
$\text{DDPG+SL}_{\text{dense, 3h}}$	72.1%	99.7%	$\mathbf{3}+3_{+0.5}$
$\text{CPO}_{\text{sparse,98\%}}$	96.8%	99.7%	$\mathbf{25}_{+0.5}$
$\text{CPO}_{\text{dense,98\%}}$	92.6%	99.4%	$\mathbf{18}_{+0.5}$
$\text{DDPG+SL}_{\text{sparse,98\%}}$	94.1%	**99.9%**	$\mathbf{21}+\mathbf{3}_{+0.5}$
$\text{DDPG+SL}_{\text{dense,98\%}}$	90.4%	**99.9%**	$\mathbf{30}+\mathbf{3}_{+0.5}$

utilized the TurtleBot3 robot model [29] within a Gazebo simulated Hospital World[1], as shown in Fig. 2.

For training the original policy π_θ, we first manually moved the robot using a joystick from two initial positions towards six designated goals, as depicted in Fig. 4. We gathered a total of 5 trajectories for each combination of initial and goal positions. we allocated a fixed duration of 0.5 hours for collecting expert demonstrations. The collected trajectories are then employed to train a two-hidden-layer policy with 256 ReLU nodes at each hidden layer using Behavioral Cloning. The collision predictive model $f(r(t))$ is assumed to be a one-hidden-layer neural network with 128 ReLU hidden nodes and a softmax final layer. For training the collision model, we allowed the robot to move randomly in the environment for a duration of 3 hours using the wander steering algorithm [30]. We finally repaired the policy using Algorithm 1 over the collected expert trajectories to ensure that $z = 0$ for all states in the training set. During the repair process, the initial values were set as $\mu_0 = 5$, $\eta = 0.001$, and $\beta = 1.5$. In experiments, we optimized the policy using the full loss term from Eq. (4) as $\text{SARP}_{\text{Lagrangian}}$, and only the quadratic penalty term ($\lambda = 0$) as $\text{SARP}_{\text{penalty}}$. Figure 4 shows the robot trajectories before (a) and after (b) repair.

We conducted a performance comparison between SARP and two safe RL methods of CPO [4], and DDPG+SL [5]. CPO considers the constraints in policy optimization and employs conditional gradient descent with line search to perform policy updates. DDPG+SL optimizes the policy with the Deep Deterministic Policy Gradient (DDPG) method and incorporates a filter to direct actions toward safety during policy exploration. For a fair comparison, we initialized the RL policies with the original policy that we trained using expert demonstrations. We also used the samples that we collected for training the collision model to pre-train the safety filter in DDPG+SL. In this study, we define the

[1]We employed the Gazebo simulated Hospital World, a publicly available environment developed by Amazon Web Services, accessible at `https://github.com/aws-robotics/aws-robomaker-small-warehouse-world.git`

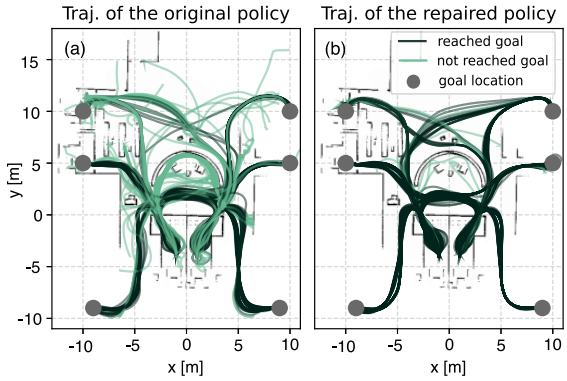

Fig. 4: Navigation in hospital: the collision avoidance of the robot improves by over 93% with SARP shown in (b) compared to the original policy depicted in (a).

simulation time (ST) as the duration that the robot interacts with the environment to collect samples for policy optimization. Therefore, in total, SARP utilized 0.5 h for collecting expert demonstrations and 3 h for samples collection used in training the collision model. Since DDPG+SL's policy and safety filter are initialized with the same samples, it used $3+0.5$ h of simulation time for initialization. In contrast, CPO only required 0.5 h for initialization, as it evaluates collisions during the training process. The remaining simulation time for DDPG+SL and CPO is dedicated to their training process. In CPO and DDPG+SL, we used two reward functions: sparse and dense. The sparse reward assigns the highest value when the robot reaches the goal and zero otherwise. The dense reward function rewards the robot for reaching the goal and penalizes it based on the change in distance to the goal, $d_g(t-1) - d_g(t)$. For comparison, we tested the optimized policies on a larger number of trajectories (200 trajectories). To assess policy performance, we defined two metrics: *Goal-reaching (GR)*, measuring the percentage of trajectories reaching goals, and *Efficacy (E)*, evaluating sample-wise constraint satisfaction by considering samples with range readings below 0.3 meters as collisions.

Evaluation. We first evaluated the accuracy of CPO, and DDPG+SL by training them for 3 h (similar simulation time SARP used for collecting collision trajectories). We terminate the RL episode 10 epochs following a collision. The results presented in Table I indicate that SARP outperforms the safe RL methods in terms of GR, showing an improvement of approximately 43% compared to 14% in RL methods. However, the RL methods demonstrate slightly better performance in terms of E, with a 2% improvement compared to SARP. We further extended the evaluation of RL methods until they reached a GR of 98% in training, which was the highest achieved by SARP as highlighted in Table I. In terms of GR, SARP outperforms CPO and DDPG+SL by 1.3%. CPO and DDPG+SL require at least an additional 15 hours compared to SARP to achieve this level of GR accuracy. In Fig. 5 (a)-(b), we demonstrate our framework's ability to enforce multiple constraints, preventing collisions and keeping velocities below 0.9 $[m/s]$. After repair, min-

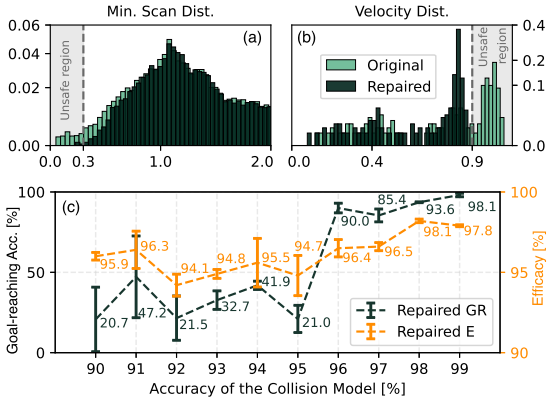

Fig. 5: Navigation in hospital: (a)-(b) minimum range sensor values and velocity distributions before and after constraint application. (c) Goal-reaching accuracy and the percentage of safe samples vs. the accuracy of the prediction model.

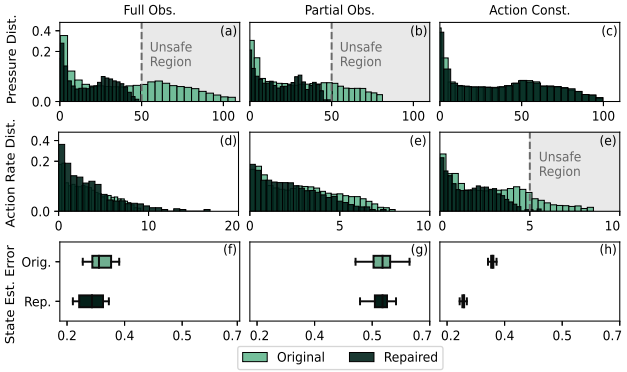

Fig. 6: Real-world walking results: (Top) distribution of pressure values, (Middle) distribution of action rate values, (Bottom) state estimation error.

imum scan values consistently exceeded 0.3, and velocities stayed below 0.9 $[m/s]$. We also investigated the relationship between GR and E versus the collision model's accuracy in Fig. 5 (c). To generate Fig. 5 (c), we conducted training on 5 networks for each level of collision accuracy. Moreover, we tested these networks with 200 trajectories each. The results demonstrate that the percentage of safe samples remains above 94% across various collision model accuracies. However, GR drops below 90% when the collision model is less than 96% accurate. While SARP improves constraint satisfaction over the original policy (74.4%), employing an inaccurate collision model leads to a deviation from the original task. Thus, SARP's accuracy can be influenced by the accuracy of the predictive model.

B. Real-world Example: Lower-leg Prosthesis Control

In this task, policy π_θ takes as input a history of the previous h observable state values at time step t, i.e., $s = [s^o(t-h+1), \cdots, s^o(t-1), s^o(t)]$. The policy then predicts the future ankle angles $a = [\alpha_a(t), \alpha_a(t+1), \cdots, \alpha_a(t+q-1)]$ of the prosthesis device for q steps ahead. To predict the system's states for the q steps ahead, we use the predictive model $f(s,a)$. Specifically, $z = [s^f(t+1), s^f(t+2), \cdots, s^f(t+q+1)]$, where the states are the angle and velocity of the upper and lower limb, and the pressure sensor insole readings, i.e., $s^f = [\alpha_{ul}, \dot{\alpha}_{ul}, \alpha_{ll}, \dot{\alpha}_{ll}, p]^T$, respectively. The states and actions of the system are shown in Fig. 3 (left) and (middle). In our experiments, we only considered the sensors located on the heel as part of system states, $p = [p_1, \cdots, p_4]^T$.

TABLE II: Policy repair statistics offline. This table reports RC: average Relative Change of state or action values with respect to their original values after repair on test data, E: average percentage of unsafe samples that are repaired (Efficacy), and SE: average relative change of prediction error after repair with respect to the original error evaluated over the originally safe state and action regions (Side Effect).

	RC (Pressure)	RC (Action)	E	SE (State)	SE (Action)
Full Obs.	**425%**	11%	**99%**	**−40%**	2%
Partial Obs.	**110%**	4.5%	**100%**	**−23%**	30%
Action Const.	13%	**118%**	**99%**	**−5%**	7%

We investigate two modes of operation: fully observable and partially observable. In the fully observable mode, the policy predicts future actions given complete sensor readings, represented as $s^o = s^f$. Conversely, in the partially observable mode, the policy receives upper and lower limb sensor readings only, denoted as $s^o = [\alpha_{ul}, \dot{\alpha}_{ul}, \alpha_{ll}, \dot{\alpha}_{ll}]^T$, while assuming the absence of pressure readings during testing.

In this experiment, we evaluate our approach on three test cases: bounding pressure for both fully observable and partially observable scenarios, and bounding the action rate solely for the fully observable case. For bounding pressure, we ensure that the pressure readings remain below 50 $[N/cm^2]$. This constraint can be defined as $\mathcal{G}_p = \{p_j(t+i) < 50 \text{ for } j = 1, \cdots, 4\}_{i=0}^{q}$. Regarding the constraint on the action rate, we aim to limit the rate of change in ankle angle to prevent abrupt actions and promote smoother transitions. We enforce a condition that restricts the absolute difference between consecutive ankle angles to be below 5 $[deg/s]$, denoted as $\mathcal{G}_{\alpha_a} = \{|\alpha_a(t+i+1) - \alpha_a(t+i)| \leq 5\}_{i=0}^{q}$. In all experiments, we specified $q = 30$ and $h = 10$.

In training, we first collected data by recording the walking gait of a healthy individual who was not using a prosthetic device under a study approved by the Institutional Review Board (IRB). We used inertial measurement units (IMUs) to capture the joint angles and pressure sensor insole to measure foot pressures. We pre-trained a two-hidden-layer policy and a one-hidden-layer predictive model with 512 ReLU nodes at each hidden layer using BC and supervised learning, respectively. We repaired the policy until all repair samples satisfied the constraints following the Algorithm 1. The first predicted ankle angle $\alpha_a(t)$ was then used as the control parameter for a PD controller on the prosthetic device. We specified $\mu_0 = 5$, $\beta = 5$ and $\eta = 0.001$.

Evaluataion. We evaluated the performance of our method in two online and offline cases. Table II shows the offline

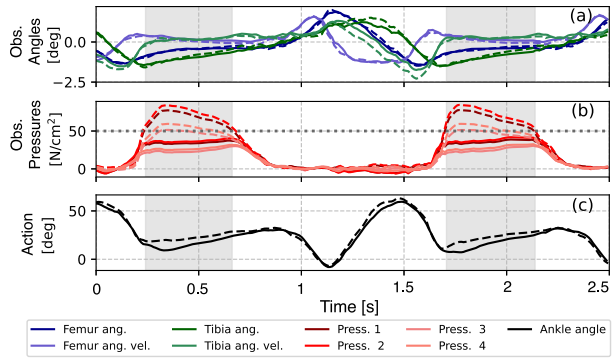

Fig. 7: Online walking signals: **fully observable case with bounded pressure values** $p \leq 50\ [N/cm^2]$. Dashed lines show the walking signals with the original model and the solid lines show the repaired signals.

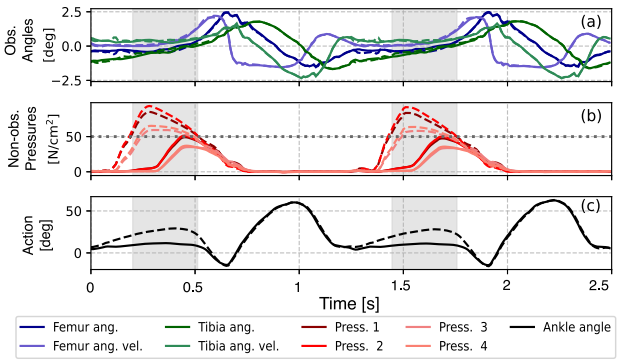

Fig. 8: Online walking signals: **partially observable case with bounded pressure values** $p \leq 50\ [N/cm^2]$. Dashed lines show the walking signals with the original model and the solid lines show the repaired signals.

evaluation of repaired versus originally trained models. We assessed the models using the test samples. In the fully observable (Full Obs.) and partially observable (Partial Obs.) test cases, the pressure values are changed by over 100% compared to the original model. This is because the repaired policy prevents the pressure values from exceeding a certain threshold. Bounding the action rate also changed the action values by 118%, as illustrated in Table II. The repair demonstrates over 99% efficacy in all models, indicating that constraint satisfaction is well-generalized to the testing samples. Finally, we evaluated the relative change in the prediction error of predictive and policy networks for the input regions that were originally safe (SE). We assessed this metric over the original testing samples. This metric is important as it measures if the repaired policy has side effects on the prediction performance of the regions that do not need to be repaired. Our results indicate that repair improved the state prediction error by up to 40% in the safe regions. However, the action error is increased in all three cases. This could be attributed to predicting actions for a future time horizon, that considers manipulating actions in advance of unsafe behavior occurrence.

Figure 6 shows the online walking results. The pressure

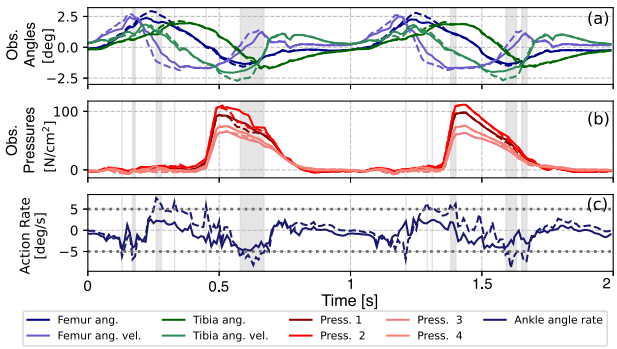

Fig. 9: Online walking signals: **fully observable case with bounded action rate** $|\Delta\alpha_a| \leq 5\ [deg/s]$. Dashed lines show the walking signals with the original model and the solid lines show the repaired signals.

distributions demonstrate that the pressure constraints are successfully satisfied for the fully observable and partially observable cases with bounded pressure values. The pressure values are bounded by $50\ [N/cm^2]$ as shown in Figures 6 (a)-(b). The real-world walking results of the action rate constrained model also show that the action rates never exceed $5\ [deg/s]$, see Fig. 6 (e). The state estimation error average is below 0.35 in fully observable and action-bounded test cases, Figures 6 (f) and (h), respectively. Nonetheless, the mean state estimation error exhibits a nearly 0.3 increase in the partially observable case, as illustrated in Fig. 6 (g). One possible explanation for this outcome is the inability of the network to capture the temporal influence of non-observable states during training as inputs. Despite this, the state estimation accuracy remains reasonably high, as evidenced by the successful pressure bounding at the heel achieved by the policy repair. Figures 7, 8, and 9 show the online walking signals for the fully observable case with bounded pressure, the partially observable test with bounded pressure, and the bounded action rate case, respectively. The pressure values are effectively constrained in both completely observable and partially observable situations, as depicted in Figures 7 and 8. In Fig. 9, the repaired model bounds the action rate to $5\ [\mathrm{deg}/s]$ with success.

C. Discussion

Our experiments in Sec. V-A showed that the accuracy of predictive model might impact the safety accuracy of the repaired policy. We also recognize that RL enables agents to explore and uncover novel solutions that may not be evident in expert demonstrations or predefined safety constraints. However, repairing policies with predictive models offers several benefits that position SARP as a reliable approach for continuous policy monitoring and repair. Firstly, it eliminates the need to explicitly estimate or learn the system's transition model, e.g., $\sigma(s(t), a(t)) = s(t+1)$. Instead, we can focus on immediately predicting the (lower-dimensional) features of interest. In our case, these variables include the physical interaction between the robot and colliders in the environment, and the bio-mechanical features. Furthermore,

predictive models enable proactive identification and mitigation of potential safety issues, promoting a proactive safety approach. It allows reasoning about the safety of after-effects and ramifications of impacts, such as assessing whether the current action would result in a collision. Finally, predictive models enhance learning efficiency by reducing the need for time-consuming simulations. This positions SARP as a suitable method to address safety concerns in scenarios where extensive exploration of safety policies is not feasible.

Controlling lower-leg prostheses serves as a compelling application for SARP in robot learning, addressing challenging dynamics of human-robot interactions and diverse safety concerns. In our current IRB-approved study, we focused on a single human subject. However, further research is required to evaluate and analyze the performance of SARP-repaired policies across multiple subjects. Moreover, including additional biomechanical indices, such as ground reaction forces or joint forces, can enhance policy safety and robustness under SARP. Finally, we tested the policy repair process for the prosthesis example with the same hyperparameters as in the hospital scenario. We observed similar efficacy outcomes after 200 epochs. In future work, we will further analyze repair sensitivity to optimization parameter selection and compare it with other algorithms.

VI. Conclusions and Future Work

Our paper presents a method that combines imitation learning with neural network repair to address safety concerns in robot learning. By leveraging predictive models, we repair policies to adhere to predefined safety constraints, enhancing safety and adaptability in robot behavior. We showcase the effectiveness of our approach in robot collision avoidance and lower-leg prosthesis control scenarios. Future work includes exploring continual learning methods to update the predictive model based on new experiences and adapt the repaired policy to changing environmental conditions and evolving safety requirements.

References

[1] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, "A survey of robots in healthcare," *Technologies*, vol. 9, no. 1, p. 8, 2021.

[2] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[3] K. Majd, G. M. Clark, T. Khandait, S. Zhou, S. Sankaranarayanan, G. Fainekos, and H. Amor, "Safe robot learning in assistive devices through neural network repair," in *6th Annual Conference on Robot Learning*. PMLR, 2022.

[4] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[5] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[6] P. Liu, D. Tateo, H. B. Ammar, and J. Peters, "Robot reinforcement learning on the constraint manifold," in *Conference on Robot Learning*. PMLR, 2022, pp. 1357–1366.

[7] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[8] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, 2023.

[9] J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," in *Robotics: Science and Systems (RSS)*, 2020.

[10] B. Chen, Z. Liu, J. Zhu, M. Xu, W. Ding, L. Li, and D. Zhao, "Context-aware safe reinforcement learning for non-stationary environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 689–10 695.

[11] Y. As, I. Usmanova, S. Curi, and A. Krause, "Constrained policy optimization via bayesian world models," in *International Conference on Learning Representations*, 2021.

[12] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[13] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in neural information processing systems*, 2018.

[14] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3612–3619, 2020.

[15] H. Yu, W. Xu, and H. Zhang, "Towards safe reinforcement learning with a safety editor policy," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2608–2621, 2022.

[16] Y. Liu, J. Ding, and X. Liu, "Ipo: Interior-point policy optimization under constraints," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 4940–4947.

[17] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[18] A. Sinitsin, V. Plokhotnyuk, D. Pyrkin, S. Popov, and A. Babenko, "Editable neural networks," *arXiv preprint arXiv:2004.00345*, 2020.

[19] X. Ren, B. Yu, H. Qi, F. Juefei-Xu, Z. Li, W. Xue, L. Ma, and J. Zhao, "Few-shot guided mix for dnn repairing," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 717–721.

[20] G. Dong, J. Sun, X. Wang, X. Wang, and T. Dai, "Towards repairing neural networks correctly," in *IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 714–725.

[21] B. Goldberger, G. Katz, Y. Adi, and J. Keshet, "Minimal modifications of deep neural networks using verification," in *23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, vol. 73, 2020, pp. 260–278.

[22] F. Fu and W. Li, "Sound and complete neural network repair with minimality and locality guarantees," in *10th International Conference on Learning Representations (ICLR)*, 2021.

[23] X. Yang, T. Yamaguchi, H.-D. Tran, B. Hoxha, T. T. Johnson, and D. Prokhorov, "Neural network repair with reachability analysis," *arXiv preprint arXiv:2108.04214*, 2021.

[24] M. Sotoudeh and A. V. Thakur, "Provable repair of deep neural networks," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 588–603.

[25] K. Majd, G. M. Clark, T. Khandait, S. Zhou, S. Sankaranarayanan, G. Fainekos, and H. Amor, "Certifiably-correct control policies for safe learning and adaptation in assistive robotics," in *Neural Information Processing Systems (NeurIPS) - Robot Learning Workshop*, 2022.

[26] J. N. S. J. Wright, "Numerical optimization," 2006.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] R. Gehlhar, M. Tucker, A. J. Young, and A. D. Ames, "A review of current state-of-the-art control methods for lower-limb powered prostheses," *Annual Reviews in Control*, 2023.

[29] "Turtlebot3 simulation model," https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/.

[30] C. W. Reynolds *et al.*, "Steering behaviors for autonomous characters," in *Game developers conference*, vol. 1999. Citeseer, pp. 763–782.