Quantifying Assistive Robustness Via the Natural-Adversarial Frontier

Anonymous Author(s)
Affiliation
Address
email

Abstract: Our ultimate goal is to build robust policies for robots that assist people. What makes this hard is that people can behave unexpectedly at test time, potentially interacting with the robot outside its training distribution and leading to failures. Even just *measuring* robustness is a challenge. Adversarial perturbations are the default, but they can paint the wrong picture: they can correspond to human motions that are unlikely to occur during natural interactions with people. A robot policy might fail under small *adversarial* perturbations but work under large *natural* perturbations. We propose that capturing robustness in these interactive settings requires constructing and analyzing *the entire natural-adversarial frontier*: the Pareto-frontier of human policies that are the best trade-offs between naturalness and low robot performance. We introduce RIGID, a method for constructing this frontier by training adversarial human policies that trade off between minimizing robot reward and acting human-like (as measured by a discriminator). On an Assistive Gym task, we use RIGID to analyze the performance of standard collaborative RL, as well as the performance of existing methods meant to increase robustness. We also compare the frontier RIGID identifies with the failures identified in expert adversarial interaction, and with naturally-occurring failures during user interaction. Overall, we find evidence that RIGID can provide a meaningful measure of robustness predictive of deployment performance, and uncover failure cases in human-robot interaction that are difficult to find manually.

Keywords: assistive robots, safety, human-robot interaction, adversarial robustness

1 Introduction

Deploying assistive robots in healthcare facilities and homes hinges crucially on their robustness, reliability, and the assurance that they can safely interact with individuals of all ages, from children to older adults. The absolute worst-case scenario would involve an errant robotic arm causing injury to humans. However, slight variations or unpredictable elements in the operational environment can drastically influence the behavior and output of learned robotic policies. Consequently, we must thoroughly examine the robot's response not just to regular situations but also to potential scenarios where unexpected events may unfold.

The complexities of stress testing robot policies, however, can be exceptionally high. For instance, verifying the robustness of a dish-loading robot in all kitchen configurations is highly costly [1]. Verifying human-robot interaction applications, in comparison, is even more challenging due to the fact that the human partner performs independent actions, changing the state that the robot's policy responds to, and potentially inducing distribution shift.

The common notion of robustness — perturb inputs x to the model — would translate here to perturbing human partner actions. But this can paint a misleading picture of the robustness of an assistive policy: while in theory any human action is possible, in practice humans won't move in arbitrary ways. A robot policy might thus fail under small *adversarial* perturbations but work under quite large *natural* deviations in human motion.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

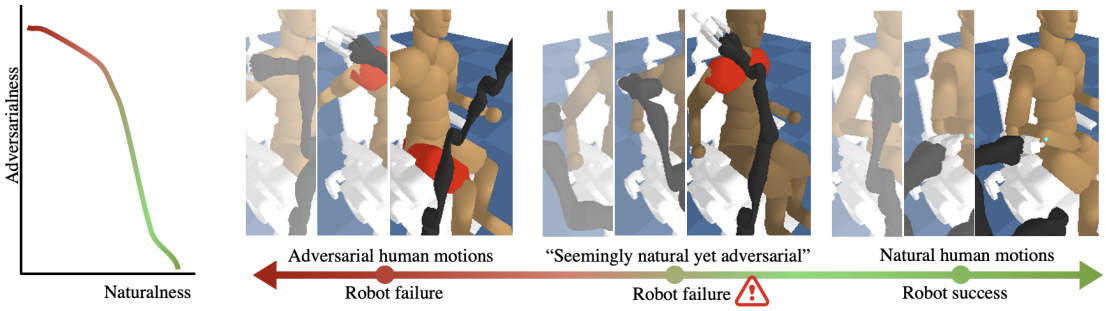


Figure 1: Our proposal is to measure assistive robustness by considering the *naturalness* of human motions, and analyzing the entire natural-adversarial frontier: points with natural human motion that lead to good robot performance (left), unnatural motions that easily break the policy (right), and potentially rather natural motions that still lead to failures (center). Looking across the full frontier paints a more useful picture of robustness than looking at a single point, and can be reduced to a scalar using the area under the curve (AUC): lower AUC is indicative of higher robustness.

We thus propose to measure robustness to *natural* variations in human motion. This begs the question of how to define "natural". We propose a generative approach: given a dataset of motions — demonstrated by the human in the target task or sampled from a trusted human model/simulator [2, 3] — we train a GAN [4, 5] and definite naturalness via the learned discriminator.

Next comes the question of *how* unnatural a motion is allowed to be that lowers the policy's performance: it is easy to break most policies if we allow motions that are very unnatural, and we might not find the interesting failure cases if we restrict the motion to be too close to what was seen in training. We thus advocate that robustness should not be assessed by looking at a single threshold, but rather by looking *at the entire natural-adversarial frontier*.

Our proposal is to measure assistive robustness by considering the naturalness of human motions, and analyzing the entire natural-adversarial frontier.

This Pareto frontier will likely contain points of poor robot performance with very unnatural human motion and points of good robot performance with natural motion. What is interesting is what happens in between — what is the area under the curve? Robust policies will have better performance as naturalness is constrained to be higher. In contrast, brittle policies can still be broken by quite natural motions, as in Fig. 1.

We introduce RIGID, a method that constructs the natural-adversarial frontier by training adversarial human policies that trade off between minimizing robot reward and acting in a human-like way, as measured by the discriminator. While in our work we use a simulated human to generate data for the discriminator, in general this data can also come from human-human or human-robot interaction in the task. We construct the frontiers for different policies — one trained to naively collaborate with the human, and others trained to be more robust (e.g. by using populations). For each, we discover natural motions (natural with regard to our simulated human) that diminish their performance. We also find that the policies prior robustness methods propose do increase robustness under RIGID too. We then use RIGID to improve robustness by fine-tuning regular RL using data points identified by RIGID. Finally, we compare RIGID with deployment with people — naive users as well as an expert who attempts to lower the robot's reward, and find that RIGID is predictive of deployment performance.

2 Related Work

Adversarial attacks and robustness in machine learning models have long been studied, with many notable examples in computer vision [6, 7, 8, 9, 10, 11]. Researchers have shown that it is easy to significantly compromise state-of-the-art computer vision models with simple manipulations of pixels [12, 13]. Such perturbations, imperceptible to human eyes, can lead to real-world safety concerns. Since then, the community has persistently come up with many attacks [14, 15, 16] and defense mechanisms [17, 18, 9]. Most of the attack methods can be characterized by their allowed perturbation set, such as bounded l2-norm, and the method of optimizing adversarial examples [19]. Robustness methods, on the other hand, often require training against adversarial examples by adding them to the

training data [7, 20, 21] as a way of performing robust optimization [22] or focus on regularization and smoothing techniques [23, 18, 24, 9].

In robotics, robust control [25] is a well-studied field that focuses on deriving safety guarantees in the presence of bounded modeling errors and disturbances [26, 27, 28, 29]. It has been applied to human-robot interaction by allowing completely adversarial human motions (the full forward reachable set), as well as a restriction to only motions that are sufficiently likely under a human model [30]. Robustness in reinforcement learning is a growing field inspired by the brittleness of RL policies [31, 32]. Prior work on adversarial attacks focuses on single-agent or two-player competitive settings [33, 34, 35, 36]. Few works have studied what it means to be robust in collaborative settings, as collaborative policies are known to be much more brittle than competitive policies, due to the importance of modeling human suboptimality [2]. Recent works have also looked at improving the generalizability in collaborative settings, which is a form of robustness against naturally occurring human distributions [37, 38, 39]. Our work is complementary, providing a way to measure robustness, and perhaps paving the way to new methods to improve it.

Generative adversarial imitation learning (GAIL) [5] is a promising method for producing motions that mimic a set of demonstrations by training a discriminator on a dataset to provide the reward for a reinforcement learning agent. Researchers have experimented with variants of GAIL and found that it leads to high-quality motions in graphics [40], locomotion [41], and even controllable natural motions [42]. In this work, we build on the existing techniques for training GAIL to generate human motions that are perceived as natural. This allows us to study assistive robustness: by optimizing for the joint objective of being both natural and adversarial, we search for a range of plausible and typical human motions that create catastrophic failures for robot policies in assistive settings.

3 Natural-Adversarial Robustness

In this section, we first introduce the robotic assistance problem in Fig. 2, where the goal is to assist a human in a partially observable setting — the human may have hidden information, such as intent or preferences, that needs to be inferred on the fly. We then formulate the problem of adversarial but natural human motion, and reduce it to the optimization of an objective trading off between the two.

Assistance as a Two-Player Dec-POMDP. Following prior work [43, 44, 37] we model an assistive task as a two-agent, finite horizon decentralized partially-observable Markov decision process (Dec-POMDP), defined by the tuple $\langle S, \alpha, A_R, A_H, \mathcal{T}, \Omega_R, \Omega_H, O, R \rangle$. Here S is the state space and A_R and A_H are the human's and the robot's action spaces, respectively. The human and the robot share a real-valued reward function $R : S \times A_R \times A_H \to \mathbb{R}$; however, we assume that the reward function is not fully observed by the robot, i.e., some of the reward function parameters (e.g., the specific goal location or objective of an assistive task) are in the hidden part of the state. $\mathcal{T}$ is the transition function where $\mathcal{T}(s' \mid s, a_R, a_H)$ is the probability of transitioning from state s to state s' given $a_R \in A_R$ and $a_H \in A_H$, Ω_R and Ω_H are the sets of observations for the robot and human, respectively, and $O : S \times A_R \times A_H \to \Omega_R \times \Omega_H$ represents the observation probabilities. We denote the horizon of the task by T.

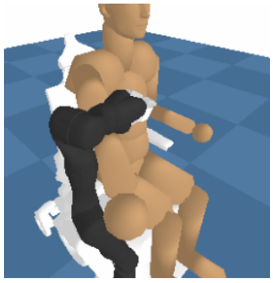


Figure 2: Assistive task: itch scratching with unknown itch locations. See appendix Sec. B for more details.

Adversarial Human Motions. Let a human trajectory denote a sequence of states and actions $\tau = (s_1, a_1^H, ..., s_T, a_T^H)$. and let function π_H be the human policy that maps from local histories of observations $\mathbf{o}_t^H = (o_1^H, \ldots, o_t^H)$ over Ω_H to a probability distribution over actions. We define for an arbitrary human policy π_H, its corresponding assistive robot policy $\pi_{R(H)}$, which is the robot policy that is optimized to collaborate best with π_H with respect to the joint reward R: $\pi_{R(H)} = \arg\max_{\pi_r} [\sum_t R(\pi_R, \pi_H)]$.[1]

We can then define an adversarial human policy $\tilde{\pi}_H$ with respect to an assistive robot policy. The policy $\tilde{\pi}_H$ minimizes the overall performance under the constraint that trajectories from the adversarial human policy must be similar to trajectories from π_H, as measured by some divergence measure $D(\tilde{\pi}_H || \pi_H)$ and adjustable coefficient δ which bounds the allowed deviation of the adversarial human

[1]For the sake of simplicity, we use a modified R to denote the expected reward of the rollouts of human-robot policy pairs as $R(\pi_R, \pi_H) := \mathbb{E}_{a_r \sim \pi_r} \mathbb{E}_{a_h \sim \pi_H} \mathbb{E}_{s_1} [\sum_t R(s^t, a_r^t, a_h^t)]$.

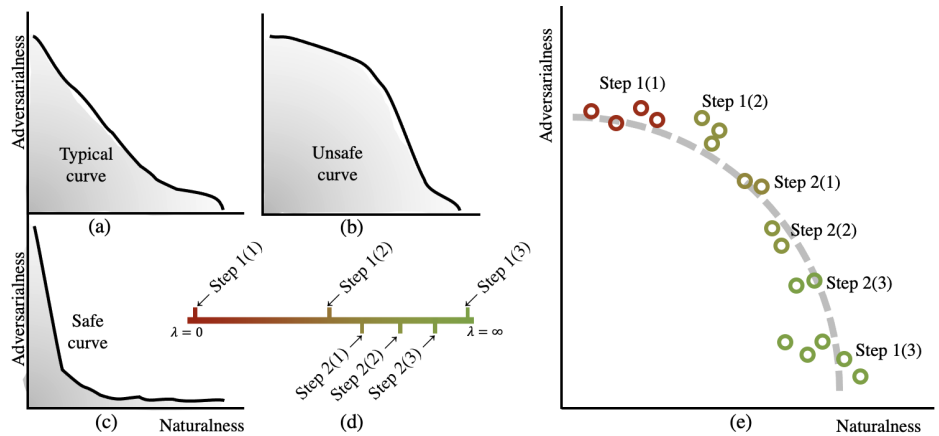


Figure 3: Left: different types of Natural-Adversarial curves, where ones with higher AUC (b) than regular (a) corresponds to more unsafe behaviours, and the ideal (c) curve looks like a sharp drop. Right: when running RIGID in Alg. 1, we first sample λ's evenly in the log space over $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ in (d). We then iteratively refine our picks by sampling more finely over selected intervals to gain better coverage of the curve in (e).

policy with respect to π_H:

$$\tilde{\pi}_H(\pi_{R(H)}, d) = \arg\min_{\pi'_H} [R(\pi_{R(H)}, \pi'_H)] \quad \text{s.t. } D(\pi'_H || \pi_H) \leq \delta \tag{1}$$

The Natural-Adversarial Tradeoff. We can further instantiate Eq. (1) by using KL divergence for D, which is commonly used to measure the distance between policies [45, 46, 3, 47]. We approximate the KL divergence with a discriminator. We train a discriminator $\mathcal{D} : \tau \to \mathbb{R}$ to distinguish between trajectories sampled from $\tilde{\pi}_H$ and π_H. More specifically, $\mathcal{D}$ assigns a score $\mathcal{D}(\pi) := \mathbb{E}_{\tau\sim\pi}[\mathcal{D}(\tau)]$ to any policy π. It is trained to assign low scores to trajectories drawn from the true human policy π_H and high scores to trajectories drawn from the adversarial policy $\tilde{\pi}_H$. In practice, we can obtain trajectories of the true π_H from human demonstrations or from a human model or simulator[2]. We consider the following discriminator

$$\mathcal{D} = \arg\min_{\mathcal{D}} \mathbb{E}_{\tau\sim\tilde{\pi}_H}\left[\log(1+\exp\{-\mathcal{D}(\tau)\})\right] + \mathbb{E}_{\tau\sim\pi_H}\left[\log(1+\exp\{\mathcal{D}(\tau)\})\right]. \tag{2}$$

As shown in prior work [46, 3], if $\mathcal{D}$ is the minimizer in Eq. (2), then $D_{\mathrm{KL}}(\tilde{\pi}_H || \pi_H) = \mathbb{E}_{\tau\sim\tilde{\pi}_H}[\mathcal{D}(\tau)]$. We denote $\mathcal{D}(\tilde{\pi}_H) := \mathbb{E}_{\tau\sim\tilde{\pi}_H}[\mathcal{D}(\tau)]$ for simplicity. We then consider the Lagrange dual function of the constrained optimization problem in Eq. (1), and replace D_{KL} using the discriminator:

$$L(\pi_R, \lambda) = \max_{\pi'_H} \left[-R(\pi_R, \pi'_H) - \lambda \cdot (D_{\mathrm{KL}}(\pi'_H || \pi_H) - \delta)\right] \tag{3}$$

$$= \max_{\pi'_H} \left[-R(\pi_R, \pi'_H) - \lambda \cdot (\mathcal{D}(\pi'_H) - \delta)\right]. \tag{4}$$

Solving for the policy that is a solution to the above Lagrangian we have

$$\tilde{\pi}_H(\pi_R, \lambda) = \arg\max_{\pi'_H} \left[-R(\pi_R, \pi'_H) - \lambda \cdot \mathcal{D}(\pi'_H)\right]. \tag{5}$$

Eq. (5) has an intuitive interpretation: we are optimizing for policy $\tilde{\pi}_H$ with respect to a combination of the inverse robot reward $-R(\pi_R, \cdot)$ (i.e., playing an adversary to the robot) and a discriminator score $-\mathcal{D}(\cdot)$ (maintaining naturalness).

Adversarial Frontier. We have reduced the constrained optimization in Eq. (1) to an unconstrained optimization in Eq. (5) that balances "adversarialness" and "naturalness" with coefficient λ. Qualitatively, λ provides a knob that we can use to trade off how adversarial and how natural we would like $\tilde{\pi}_H$ to be. Setting $\lambda \to \infty$ results in a policy, $\tilde{\pi}_H$, that closely resembles π_H and yields a high reward. On the other hand, setting $\lambda = 0$ leads to a policy, $\tilde{\pi}_H$, that is purely adversarial and

[2]Typically, robot policies are trained via RL as a best response to a human model [2]; offline RL [48] is also possible and bypasses a human model, in which case the discriminator can be trained directly on the data offline RL uses.

causes harm to the assistive task by inverting the environment reward. By selecting $\lambda \in [0, \infty)$, we arrive at a spectrum of human motions that interpolate between adversarial and natural.

Given these definitions, we can in principle compute the naturalness and adversarialness of all possible human motions, and plot them in a 2D coordinate system. This motivates the Natural-Adversarial curve — the Pareto frontier of non-dominated policies. These are policies that, compared to any other ones, have either a better adversarial score, or a better naturalness score, or both.

4 Computing the Natural-Adversarial Frontier

In this section, we propose a practical method that we call RIGID that can help us obtain the Adversarial Frontier in Sec. 3. Having motivated the concept of an adversarial human policy $\tilde{\pi}_H(\pi_R, \lambda)$ and the naturalness parameter, λ, in Sec. 3, we now discuss the Natural-Adversarial Pareto frontier (Sec. 4.1). We then introduce an algorithm, RIGID (Sec. 4.2), that efficiently approximates the natural-adversarial curve.

4.1 Connecting The Dots: the Natural-Adversarial Curve

Points Along the Frontier: To solve for Eq. (5), we start with sampled trajectories from π_H (in our experiments, a human model; in general, this can also be data collected from human-human or human-robot interaction). For a specific λ, we adapt GAIL [5] to our setting: we interleave training the adversarial policy $\tilde{\pi}_H$ and the discriminator $\mathcal{D}$ by iteratively (a) doing one step of policy optimization on Eq. (5) to adapt the policy and (b) doing one gradient step on Eq. (2) to adapt the discriminator. In practice, we use PPO [49, 50] as our policy optimization algorithm, and we use LS-GAN [51] with noise annealing to ensure the stability of the training process.

To plot the resulting $\tilde{\pi}_H$ in the natural-adversarial coordinate, we compute naturalness (x coordinate) by sampling trajectories $\tau \sim \tilde{\pi}_H$, and calculate the mean likelyhood of $\mathcal{D}(\tau) < 0$. This is equivalent to having the discriminator classify the trajectories as from the original human model. We define adversarialness (y coordinate) as negative robot reward normalized to [0, 1]. See appendix Sec. D.

Approximating the Curve: Once we sample a sufficient number of λs (see Sec. 4.2 on how to do this) from $[0, \infty)$, we connect the outermost points. The resulting curve characterizes the frontier of adversarial human motions under different levels of naturalness constraints (Eq. (5)) — or different perturbation sets (Eq. (1)). Intuitively, under the same naturalness value, lower adversarialness corresponds to more robust robot policies. Fig. 3 shows three different hypothetical curves, visualizing how the shape of the curve connects to the robustness of the policy.

The AUC Score as a Scalar Robustness Metric: Given a Natural-Adversarial curve, we compute the Area Under Curve (AUC) to quantitatively measure robustness as a single scalar. Lower AUC generally means stronger robustness, although there may be outlier human policies that can trigger unsafe behaviours, and a full plot of the curve is still necessary.

Algorithm 1 Procedure: RIGID

Require:
Maximum refinements d, samples per round k
Upper and lower bounds on λ: $\lambda_{\max}$, $\lambda_{\min}$
procedure RIGID ($\lambda_{\min}$, $\lambda_{\max}$, d)
$\Lambda_{\text{all}} = []$, $S_{\text{nat}} = []$, $S_{\text{adv}} = []$
$\lambda'_{\min} = \lambda_{\min}$, $\lambda'_{\max} = \lambda_{\max}$
for $i = 1, \ldots, d$ **do**
  Find k evenly spaced $e_1 \ldots e_k$ in $[\log \lambda'_{\min}, \log \lambda'_{\max}]$
  for $j = 1, \ldots, k$ **do**
    $\lambda_j = \exp(e_j)$
    Λ_{all}.append(λ_j)
    Compute $\tilde{\pi}_H$ using Eq. (5) and λ_j
    Compute nat_j and adv_j for $\tilde{\pi}_H(\lambda_j)$
    S_{nat}.append(nat_j), S_{adv}.append(adv_j)
  end for
  $\lambda'_{\min}$, $\lambda'_{\max}$ = LARGEST-JUMP(S_{nat},Λ_{all})
end for
return S_{nat}, S_{adv}
end procedure

4.2 Sampling Useful Trade-Offs

When we gradually increase λ, the resulting adversarial human motion often goes through mode changes, which lead to sudden increases in naturalness. Motivated by this, we need to select $\lambda \in [0, \infty)$ in a non-uniform manner, by procedurally expanding between the two λ's where naturalness increases

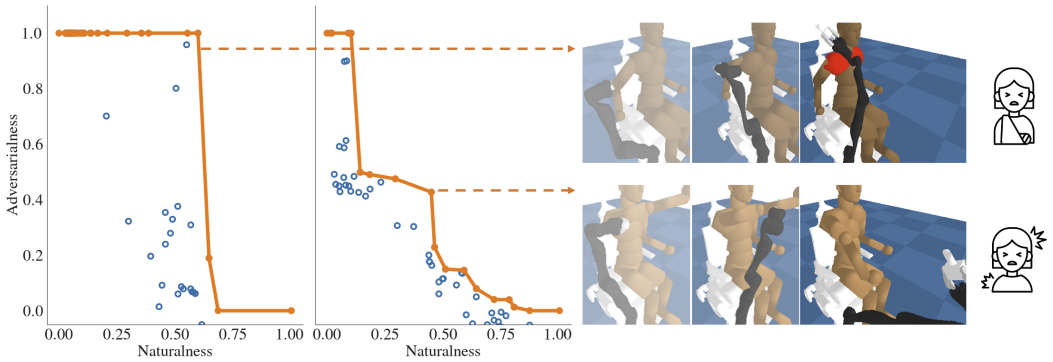


Figure 4: Natural-Adversarial curves of two different robotic policies trained using vanilla RL, with random seed differences. Every point corresponds to a motion policy found by RIGID for a particular λ value. We highlight the frontier in orange. The visualization shows how points on the frontier correspond to failure cases.

the most. We do this through an iterative refinement approach by finding the largest gap in naturalness and increasing the sampling density there.

Our algorithm's pseudocode is shown in Alg. 1. The lists S_{nat} and S_{adv} keep track of the corresponding naturalness and adversarialness values of points along the Pareto frontier. We initialize $\lambda_{\min} > 0$ to be the smallest λ we consider, and $\lambda_{\max}$ a sufficiently large number. We denote all the λ considered so far as Λ_{all}. The procedure LARGEST-JUMP looks at all λ selected so far, along with their naturalness, and selects two λ values between which the naturalness changed the most. See appendix Sec. A for the pseudo-code.

5 Experiments

In this section, we use RIGID to analyze vanilla RL policies trained against a human model and uncover possible natural (according to our simulator) motions that trigger failures. We then examine whether the RIGID frontiers are predictive of deployment performance: we test these policies with (a) end users and (b) an expert who adversarially attempts to get the robot to fail. We then plot the resulting behaviors relative to the RIGID-computed frontier. We find that RIGID is able to find points that are just as natural, but more adversarial — the caveat is that naturalness here is only as good as the human model we used in the experiments. This could certainly be improved, which we think would further improve the utility of RIGID. Next, we analyze previous algorithms proposed for robustifying RL policies, and compare them according to RIGID to show that RIGID does help differentiate between more or less robust policies. We also show that RL can be more robust by training with the RIGID-identified behaviors.

5.1 How robust is Vanilla RL?

First, we study whether assistive robot policies trained with SOTA reinforcement learning algorithms are robust. To do this, we first generate synthetic human policies by jointly optimizing for human and robot policies following [52]. We keep and freeze the resulting human policy as the synthetic human that we use, and train an assistive robot policy using the PPO algorithm to assist the synthetic human. The resulting robot policies achieve nearly 100% success rate with the training human model. We then use RIGID to come up with natural and adversarial human policies to attack this robot policy. See appendix Sec. D for further details.

In Fig. 4, we visualize the resulting human policies from the RIGID algorithm as points and connect the dots on the outermost boundary to form a Natural-Adversarial frontier. Visualizations of points with naturalness values in the range $[0.1, 0.8]$ show abundant failure cases — the adversarial human policy moves naturally (according to the model), yet causes the robot to fail. We show two examples in Fig. 4 with the dashed arrow. See appendix Sec. E for more visualizations.

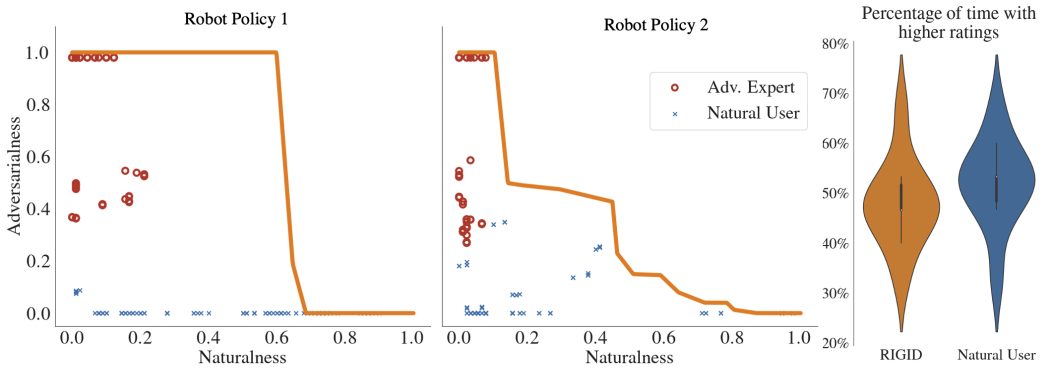


Figure 5: The RIGID-identified frontier for two policies. We find that RIGID pareto-dominates natural user and adversarial expert behaviors. When taking two behaviors with the same naturalness score — a user's and RIGID's — RIGID finds more adversarial solutions. On the right, we see that users indeed do not find a large difference between these two behaviors in "naturalness" (similarity to human training data).

5.2 How do Natural-Adversarial curves from RIGID align with user judgments?

While we can verify that points on the Natural-Adversarial curve are indeed failure cases of the robot policy, one may still question whether this curve is *faithful* and *exhaustive*. Here, *faithful* means that the trajectories deemed natural by the curve are judged as natural to the same extent by human beings; *exhaustive* means that this curve encloses all the possible failure cases that one can produce.

We perform a two-part user study based on a virtual reality (VR) assistive gym plugin [53] to verify *exhaustiveness*. First, we recruit eight novice users who do not have prior experience interacting with the robot in VR. We inform them about the task (objective and physical constraints) and have them watch a few episodes of the robot performing successfully. The robot then assists the users. This tests each robot policy's performance under regular deployment conditions, but it does not stress-test the system to emulate what might happen under a wider range of users and over prolonged deployment. To stress test the system, we have an expert (one of the authors) act adversarially to cause the robot to fail (collide). If the curve produced by the RIGID algorithm is *exhaustive*, trajectories from both the regular users as well as the expert should lie underneath the curve, which is indeed the case (Fig. 5): *points on the RIGID curve Pareto-dominate benign user and even expert adversarial data*.

We also see that for these two policies, the AUC is smaller for the second, so we expect it to be more robust. Indeed, in both user and expert data, the robot performance is higher for the second policy (lower adversarialness), suggesting *RIGID correctly identified which policy is more robust*. Anecdotally, the expert reported that the first policy was easier to break by simply extending their arm, which sent the policy into an unstable rotating behavior.

These results provide evidence that RIGID can find more adversarial behaviors for the same naturalness value. To test whether the curves are *faithful*, we ask 15 novice users to judge whether the RIGID behavior is indeed just as natural. To separate the discriminator idea (which is fundamental to RIGID) from the quality of the data (which can change), we asked users to rate not which motion is more natural, but which motion is closer to the reference data. The plot on the right of Fig. 5 shows that both get chosen with similar percentages, indicating similar distance from the reference data.

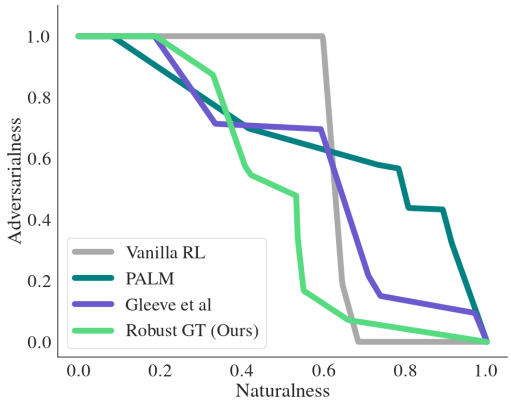
5.3 Do existing robust RL methods improve robustness? Can training with RIGID examples help?

In this section, we leverage RIGID to efficiently find Natural-Adversarial curves for different robust robot learning methods, allowing us to compare their assistive robustness.

In Fig. 5.3, we compare the following methods: (1) Gleeve et al [54] use human-policy randomization during the co-optimization phase to improve the robustness of the robot, (2) PALM [37] leverages a distribution of different human behaviors to learn a latent representation that enables better generalizations. (3) We also study an oracle method called Robust GT, where we fine-tune Vanilla RL on failure cases from the Natural-Adversarial curve.

We notice that while existing robustness methods improve over Vanilla RL in terms of Assistive Robustness AUC, they still have visible failure cases (see appendix Sec. E for visualizations). We also

plot the performance of different robust policies on the original human versus the AUC. The robust optimization method from Gleeve et al [54] improves AUC at the cost of worse performance when assisting more natural human policies. For PALM [37], we also observe this performance-robustness trade-off. We find that Robust GT, which is trained on adversarial human policies found from RIGID achieves the best reduction in AUC while also achieving good performance when assisting natural human policies.



Method	AUC (↓)	Success (↑)
Vanilla RL	0.630	1.0
Gleeve et al [54]	0.584	0.83
PALM [37]	0.668	0.95
Robust GT (Ours)	**0.473**	**1.0**

Figure 6: Left: Natural-Adversarial curves of different methods. Right: We compute the Area Under Curve (AUC) of the Natural-Adversarial curves and the success rate of robot policies trained with different methods.

6 Limitations

One of our main limitations is the discriminator in the GAN that we use to detect natural perturbations. Since we use a state-based discriminator, we are only able to detect human poses that are out of distribution. Ideally, we can have a temporal trajectory-based discriminator that detects unnatural human state transitions, such as sudden jerky motions. Training such sequential discriminator, however, imposes new challenges on GAN training and is beyond the scope of our focus.

The second limitation is we rely on synthetic humans generated by human-robot co-optimizations [52, 55]. While such human agents perform reasonable movements (see visualizations in appendix Sec. B) and serve as a sufficient test bed for our studies, future work should incorporate real-world human data to create more human-like simulated agents. Future work should explore using Behaviour Cloning or Offline RL [48] to better incorporate human data into our framework. It is also worth noting that in our studies, we use data from simulated humans as the canonical datasets on which we train GAN and measure "naturalness". While this may seem limiting, as we have only tested on our simulated humans, we think the framework is applicable to human-robot applications where there typically exists a canonical way of interaction (i.e. dressing requires the human to extend their arm to initiate the contact).

Last but not least, our proposed algorithm RIGID is time-consuming and computationally expensive. Computing a Natural-Adversarial curve typically requires training 50 to 100 RL policies. The iterative refinement process currently does not enable parallelizing different refinement cycles in Alg. 1, meaning that we must wait for several cycles of RL to finish sequentially. This makes it prohibitively slow to perform robust training methods in Sec. 5.3 for more than one iteration. We believe that Quality-Diversity methods [56, 57] as well as computation sharing (i.e. shared discriminator) between different RIGID trials are promising directions to drastically speed up the RIGID algorithm.

7 Conclusions

We propose Assistive Robustness as a measure for evaluating the robustness of robot policies in assistive settings. Verified by user studies, we show that the Natural-Adversarial curve effectively represents to what extent the robot remains safe under plausible human perturbations. Our proposed algorithm RIGID is an initial attempt at effectively computing the curve. While we study physical interaction in healthcare as the main application in this work, we believe our framework can apply to general Human-Robot collaborative settings. We are excited for proposing a framework and tools to facilitate research safety in assistive and collaborative settings, as more sophisticated robotic applications are being deployed in the real world.

References

[1] R. Tedrake. Tri taking on the hard problems in manipulation research toward making human-assist robots reliable and robust, URL.

[2] M. Carroll, R. Shah, M. K. Ho, T. L. Griffiths, S. A. Seshia, P. Abbeel, and A. D. Dragan. On the utility of learning about humans for human-ai coordination. *CoRR*, abs/1910.05789, 2019. URL http://arxiv.org/abs/1910.05789.

[3] C. Laidlaw and A. Dragan. The boltzmann policy distribution: Accounting for systematic suboptimality in human models, 2022.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio. Generative adversarial networks, 1–9. *arXiv preprint arXiv:1406.2661*, 2014.

[5] J. Ho and S. Ermon. Generative adversarial imitation learning, 2016.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[8] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[9] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

[10] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[11] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[12] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[13] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[14] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel. Adversarial attacks on neural network policies. *arXiv*, 2017. URL https://arxiv.org/abs/1702.02284.

[15] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

[16] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[17] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *CoRR*, abs/1710.10766, 2017. URL http://arxiv.org/abs/1710.10766.

[18] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[19] Z. Kolter and A. Madry. Adversarial robustness - theory and practice, URL.

[20] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[21] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[22] U. Shaham, Y. Yamada, and S. Negahban. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*, 2015.

[23] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[24] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[25] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 2007.

[26] J.-J. E. Slotine. The robust control of robot manipulators. *The International Journal of Robotics Research*, 4(2):49–64, 1985.

[27] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981. doi:10.1109/TAC.1981.1102603.

[28] M. Zeinali and L. Notash. Adaptive sliding mode control with uncertainty estimator for robot manipulators. *Mechanism and Machine Theory*, 45(1):80–90, 2010. ISSN 0094-114X. doi: https://doi.org/10.1016/j.mechmachtheory.2009.08.003. URL https://www.sciencedirect.com/science/article/pii/S0094114X09001530.

[29] E. Sariyildiz, R. Oboe, and K. Ohnishi. Disturbance observer-based robust control and its applications: 35th anniversary overview. *IEEE Transactions on Industrial Electronics*, 67(3): 2042–2053, 2019.

[30] A. Bajcsy, S. Bansal, E. Ratner, C. J. Tomlin, and A. D. Dragan. A robust control framework for human motion prediction. *IEEE Robotics and Automation Letters*, 6(1):24–31, 2020.

[31] C. Zhang, O. Vinyals, R. Munos, and S. Bengio. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018.

[32] A. Gleave, M. Dennis, N. Kant, C. Wild, S. Levine, and S. Russell. Adversarial policies: Attacking deep reinforcement learning. *CoRR*, abs/1905.10615, 2019. URL http://arxiv.org/abs/1905.10615.

[33] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel. Adversarial attacks on neural network policies. *CoRR*, abs/1702.02284, 2017. URL http://arxiv.org/abs/1702.02284.

[34] A. Mandlekar, Y. Zhu, A. Garg, L. Fei-Fei, and S. Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939. IEEE, 2017.

[35] K. Mo, W. Tang, J. Li, and X. Yuan. Attacking deep reinforcement learning with decoupled adversarial policy. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[36] T. T. Wang, A. Gleave, N. Belrose, T. Tseng, J. Miller, M. D. Dennis, Y. Duan, V. Pogrebniak, S. Levine, and S. Russell. Adversarial policies beat professional-level go ais. arXiv:2211.00241v1 [cs.LG], 2022.

[37] J. Z.-Y. He, Z. Erickson, D. S. Brown, A. Raghunathan, and A. Dragan. Learning representations that enable generalization in assistive tasks. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 2105–2114. PMLR, 14–18 Dec 2023.

[38] D. Strouse, K. R. McKee, M. M. Botvinick, E. Hughes, and R. Everett. Collaborating with humans without human data. *CoRR*, abs/2110.08176, 2021. URL https://arxiv.org/abs/2110.08176.

[39] A. Lupu, B. Cui, H. Hu, and J. Foerster. Trajectory diversity for zero-shot coordination. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7204–7213. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lupu21a.html.

[40] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *CoRR*, abs/1804.02717, 2018. URL http://arxiv.org/abs/1804.02717.

[41] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions, 2022.

[42] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. AMP: adversarial motion priors for stylized physics-based character control. *CoRR*, abs/2104.02180, 2021. URL https://arxiv.org/abs/2104.02180.

[43] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[44] T. Taha, J. V. Miró, and G. Dissanayake. A pomdp framework for modelling human interaction with assistive robots. In *2011 IEEE International Conference on Robotics and Automation*, pages 544–549. IEEE, 2011.

[45] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[46] F. Huszár. Variational inference using implicit distributions, 2017.

[47] J. Tien, J. Z.-Y. He, Z. Erickson, A. Dragan, and D. S. Brown. Causal confusion and reward misidentification in preference-based reward learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[48] J. Hong, A. Dragan, and S. Levine. Learning to influence human behavior with offline reinforcement learning. *arXiv preprint arXiv:2303.02265*, 2023.

[49] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[50] I. Kostrikov. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

[51] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks, 2017.

[52] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu. Learning to collaborate from simulation for robot-assisted dressing. *IEEE Robotics and Automation Letters*, 5(2): 2746–2753, 2020.

[53] Z. Erickson, Y. Gu, and C. C. Kemp. Assistive VR gym: Interactions with real people to improve virtual assistive robots. *CoRR*, abs/2007.04959, 2020. URL https://arxiv.org/abs/2007.04959.

[54] P. Czempin and A. Gleave. Reducing exploitability with population based training. In *New Frontiers in Adversarial Machine Learning Workshop at ICML*, 2022.

[55] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp. Assistive gym: A physics simulation framework for assistive robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10169–10176. IEEE, 2020.

[56] M. C. Fontaine and S. Nikolaidis. Differentiable quality diversity. *CoRR*, abs/2106.03894, 2021. URL https://arxiv.org/abs/2106.03894.

[57] J. Mouret and J. Clune. Illuminating search spaces by mapping elites. *CoRR*, abs/1504.04909, 2015. URL http://arxiv.org/abs/1504.04909.