

From Demonstrations to Safe Deployment: Path-Consistent Safety Filtering for Diffusion Policies

Ralf Römer^{*,1}, Julian Balletshofer^{*,1}, Jakob Thumm², Marco Pavone²,
Angela P. Schoellig¹, and Matthias Althoff¹

Abstract—Diffusion policies (DPs) achieve state-of-the-art performance on complex manipulation tasks by learning from large-scale demonstration datasets, often spanning multiple embodiments and environments. However, they cannot guarantee safe behavior, requiring external safety mechanisms. These, however, alter actions in ways unseen during training, causing unpredictable behavior and performance degradation. To address these problems, we propose path-consistent safety filtering (PACS) for DPs. Our approach performs path-consistent braking on a trajectory computed from the sequence of generated actions. In this way, we keep the execution consistent with the training distribution of the policy, maintaining the learned, task-completing behavior. To enable real-time deployment and handle uncertainties, we verify safety using set-based reachability analysis. Our experimental evaluation in simulation and on three challenging real-world human-robot interaction tasks shows that PACS (a) provides formal safety guarantees in dynamic environments, (b) preserves task success rates, and (c) outperforms reactive safety approaches, such as control barrier functions, by up to 68% in terms of task success. Videos are available at our project website: tum-lsy.github.io/pacs.

I. INTRODUCTION

One of the main goals of robotics is to reduce the mental and physical burden on human workers by automating tedious and repetitive tasks. Recently, imitation learning using diffusion models [1] and flow matching [2] has enabled robots to successfully perform these complex, long-horizon manipulation tasks [3], [4]. These advances have been reinforced by the availability of large-scale demonstration datasets [5] and the emergence of vision-language-action models [6], [7]. Diffusion policies (DPs), i.e., robot policies parameterized by generative diffusion or flow-matching models, can perform tasks in dynamic environments [8], [9]. However, since DPs are black-box models, they lack safety guarantees for avoiding collisions or safely interacting with dynamic objects. This prevents their deployment in human-centric environments, where safety has to be formally guaranteed.

Many works have shown how safety can be ensured for learning-based policies during deployment [10], [11], e.g., using control barrier functions [12], [13] or predictive safety filters [14], [15]. Most of these methods are reactive safety mechanisms that satisfy safety constraints by changing the

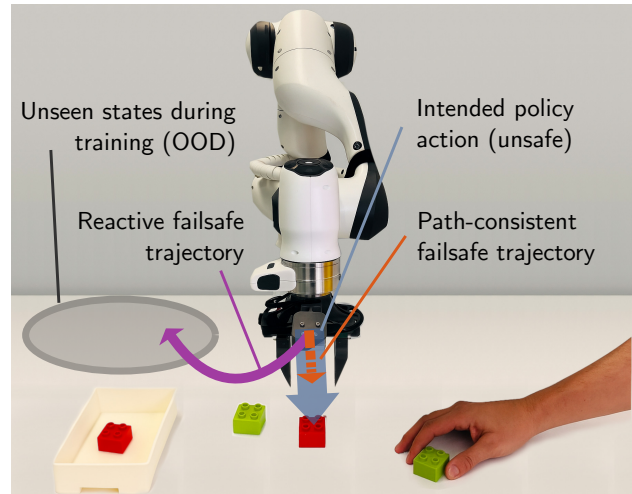


Fig. 1. Deploying DPs in dynamic environments with moving objects requires safeguarding mechanisms, as the intended policy actions may be unsafe. Reactive strategies, such as control barrier functions, often drive the agent into out-of-distribution (OOD) states not seen during training, leading to unpredictable behavior. We propose that safety mechanisms for DPs should remain consistent with the intended path of the robot to avoid out-of-distribution states and preserve high task success rates.

path of the robot away from the dynamic object. However, the vast amount of available demonstrations [5] is not recorded with any of such safety mechanisms in place. Therefore, as illustrated in Figure 1, reactive safety mechanisms often drive the system away from the demonstration data distribution into unseen states. DPs are particularly prone to errors in these out-of-distribution cases, increasing the risk of task failure [16]–[18]. To avoid this out-of-distribution situation, the interventions of a safety mechanism for DPs should ideally keep the system close to the desired actions of the policy.

We achieve this by proposing a Path-consistent safety filter (PACS) for DPs. The core idea is to use path-consistent braking to guarantee safety while keeping the system on its intended path. To this end, PACS translates the entire action sequence (chunk) generated by the policy into an intended trajectory, along which the robot is slowed down or stopped. We use set-based reachability analysis [19] to verify safety constraints, enabling real-time deployment at 1 kHz.

In summary, our main contributions are:

- We perform the first provably safe deployment of DPs for human-robot interaction (HRI) tasks, representing safety-critical and challenging examples of dynamic

^{*}Equal contribution.

¹ Department of Computer Engineering, Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, Germany. {[ralf.roemer](mailto:ralf.roemer@tum.de), [julian.balletshofer](mailto:julian.balletshofer@tum.de), [angela.schoellig](mailto:angela.schoellig@tum.de), [althoff](mailto:althoff@tum.de)}@tum.de

² Department of Aeronautics and Astronautics, Stanford University, USA. {[thumm](mailto:thumm@stanford.edu), [pavone](mailto:pavone@stanford.edu)}@stanford.edu

real-world environments.

- We demonstrate that by avoiding OOD states, PACS achieves significantly higher task success rates than reactive safety filters, both in simulated (68% better) and hardware experiments (37% better).
- We demonstrate that an intermediate trajectory generation from action chunks lead to a 28% improvement in task success rates compared to sequentially treating each action individually.

II. RELATED WORK

a) Safety Filters in Robotics: Deploying autonomous agents in safety-critical applications requires mechanisms to prevent harmful behavior. A common approach is to apply safety filters, which have been extensively reviewed in recent surveys [10], [11], [20], [21]. Safety filters typically verify whether a failsafe action exists that can bring the robot to a safe state after executing the action proposed by the policy [21]. Several techniques follow this approach: Model predictive safety filters [14], [22] and control barrier functions [12], [13], [23], [24] formulate an optimization problem to compute the safe control input closest to the desired input. In contrast, methods based on reachability analysis [25]–[31] ensure safety by keeping available a provably correct failsafe trajectory. These failsafe trajectories can be path-consistent, stopping the robot along its intended path [29], [30], [32], or non-path-consistent, guiding the robot away from unsafe regions.

b) Safety of Diffusion Policies: Compared to more classical programming-by-demonstration approaches [33], the stochastic and end-to-end nature of diffusion policies makes them particularly prone to unsafe and erratic behavior, especially in unseen situations [16]–[18]. One way to address this problem is by adding an additional safety layer to the learned policy, e.g., using control barrier functions [34]. Another example is RAIL [35], which maintains a failsafe plan based on reachability analysis, defined as a stopping motion that stays as close as possible to the nominal trajectory. While theoretically well-grounded, such post-processing methods only consider static constraints so far and significantly distort the action distribution, leading to a notable drop in task performance [35]. To address this problem, several works directly incorporate safety constraints into the iterative action generation (denoising) process of diffusion- and flow-based policies. Adding cost gradients [36] or classifier-free guidance [37] can steer the generated action sequence towards safe regions, but these methods do not guarantee hard constraint satisfaction. As an alternative, iterative projections to a safe set [38]–[40] or control barrier functions [41], [42] have been injected into the denoising process to provide certain safety guarantees. However, these approaches still change the policy, and their high computational costs limit their applicability to low-dimensional systems [38], [39] or offline planning [40]–[42]. In summary, real-time safeguarding of DPs in dynamic environments without compromising task success remains an open problem.

III. PROBLEM STATEMENT

We aim to ensure a safe deployment of DPs when operating in close proximity to or interacting with dynamic objects, including humans. More specifically, we consider a robotic system with state $\mathbf{x}(t) \in \mathcal{X}$ and input $\mathbf{u}(t) \in \mathcal{U}$ at time t that should perform a given task while adhering to a set of safety constraints. We assume that the state \mathbf{x} of the system contains the pose of dynamic objects up to a bounded measurement error and that the objects have a bounded input set, e.g., a maximal velocity. We assume the availability of a set of trajectories demonstrated by experts $\mathcal{D} = \{\mathbf{o}_k, \mathbf{a}_k\}_{k=0}^K$ in the form of observation-action pairs. The observations $\mathbf{o} \in \mathcal{O}$ may include robot joint angles, camera images, language commands, and other environmental information. The action $\mathbf{a} \in \mathcal{A}$ is a high-level goal that the robot should reach, e.g., a desired joint configuration. From the expert demonstrations, a DP $\pi : \mathcal{A}^H \times \mathcal{O} \rightarrow \mathbb{R}_+$ is trained that returns a distribution over action chunks $\mathbf{A} \in \mathcal{A}^H$ of length H . In this work, we aim to design a controller $\mathbf{k} : \mathcal{X} \times \mathcal{A}^H \times \mathbb{R} \rightarrow \mathcal{U}$ that takes as input the latest action chunk \mathbf{A} generated by the policy and returns a safe control input $\mathbf{u}_{\text{safe}}(t)$.

IV. PRELIMINARIES

This section describes the technical background on DPs and reachability-based safety verification. Throughout the paper, we use (\cdot) as a placeholder for the arguments of a probability distribution or function.

A. Imitation Learning with Diffusion Policies

The demonstration data \mathcal{D} encodes the desired behavior for solving the task. A DP π is trained to approximate the conditional action distribution in the demonstrations [3]. Once trained, DPs and VLAs generate new actions by iteratively transforming a random noise sample into the learned target distribution [1], [2]. Most DPs return a chunk of H consecutive actions $\mathbf{A}_k = (\mathbf{a}_k, \mathbf{a}_{k+1}, \dots, \mathbf{a}_{k+H-1})$ instead of only the next action to improve the temporal consistency and execution speed [3], [43]. During deployment, we can iteratively sample new action chunks $\mathbf{A}_k \sim \pi(\cdot | \mathbf{o}_k)$, apply the first $h \leq H$ actions $\mathbf{a}_{k:k+h-1}$ to the system for a time Δt , respectively, and replan after a time $h\Delta t$ has passed. This general formulation of imitation learning policies encompasses state-of-the-art generative single-task policies [3], [4] and generalist vision-language-action models [6], [7].

B. Safety Constraints for Dynamic Objects

The robot and the objects are considered as dynamic systems, and we define the trajectory of a system at time t for a given initial state $\mathbf{x}_0 \in \mathcal{X}_0$ and a possible input trajectory $\mathbf{u}(\cdot)$ by $\chi(t; \mathbf{x}_0; \mathbf{u}(\cdot))$. Following [19, p. 5], we formally introduce the reachable set $\mathcal{R}(t)$ of a system at time t for a compact set of possible inputs \mathcal{U} by

$$\mathcal{R}(t) := \{\chi(t; \mathbf{x}_0; \mathbf{u}(\cdot)) \mid \mathbf{x}_0 \in \mathcal{X}_0, \forall t : \mathbf{u}(t) \in \mathcal{U}\}, \quad (1)$$

and denote all states that are reachable during a time interval as $\mathcal{R}([t_a, t_b]) = \bigcup_{t \in [t_a, t_b]} \mathcal{R}(t)$. We denote the *reachable*

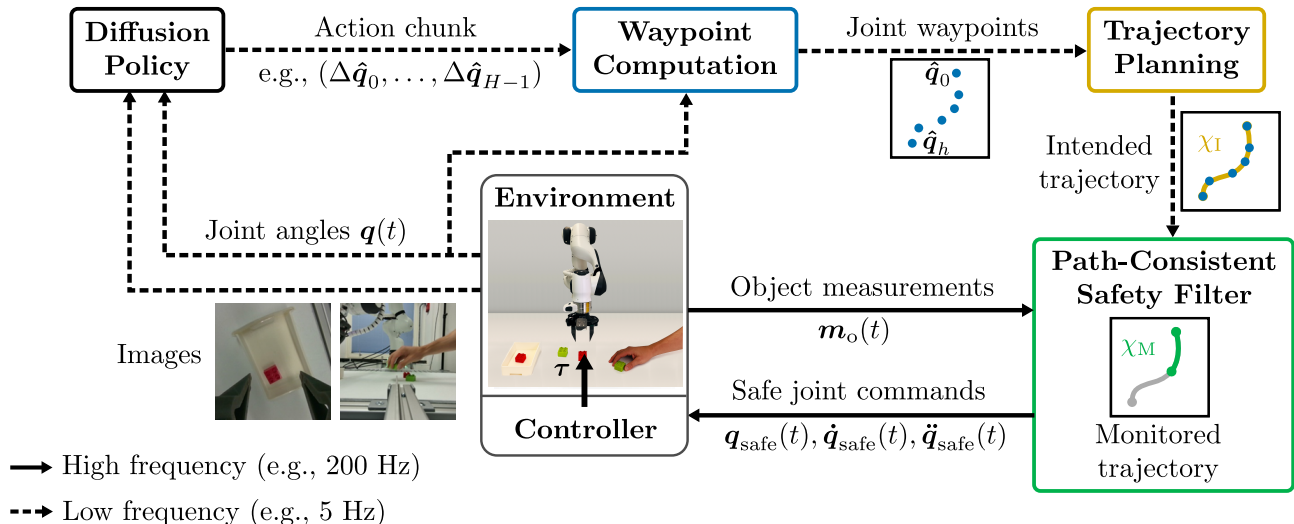


Fig. 2. System overview of our proposed path-consistent safety filter (PACS). The policy, conditioned on visual observations and proprioceptive inputs, generates action chunks that are transformed into a sequence of desired waypoints. From these waypoints, we compute a kinematically and dynamically feasible intended trajectory. PACS continuously monitors this trajectory and applies high-frequency safety filtering using reachability analysis to enforce task-specific safety constraints (e.g., collision avoidance or impact force limits).

occupancy by $\mathcal{C}(t)$. For the reachable occupancy computation of the robot $\mathcal{C}_{\text{robot}}(t)$, we assume a bounded tracking error of the desired trajectory. This assumption could be replaced by applying more advanced methods as presented in [44]. To compute the reachable occupancies of dynamic objects $\mathcal{C}_{\text{object}}(t)$, we assume access to measurements of positions of objects with bounded measurement errors. Moreover, we consider velocity and acceleration limits, and further include sensor delay and bounded measurement errors on position and velocity, following the approach in [19].

Following prior work [19], [45], [46], we define a collision between an object and the robot if their reachable occupancies $\mathcal{C}_{\text{object}}(t)$ and $\mathcal{C}_{\text{robot}}(t)$ intersect, i.e., a collision event is defined as

$$c_{\text{coll}}(t) := \mathcal{C}_{\text{object}}(t) \cap \mathcal{C}_{\text{robot}}(t) \neq \emptyset. \quad (2)$$

Depending on the task and object, collisions should either be avoided altogether or restricted to non-harmful, low-force interactions. Accordingly, we define two types of safety constraints in accordance with ISO/TS 15066:2016. Power and force limiting (PFL) targets human-robot collaboration with close interaction and allows contacts with humans as long as the kinetic energy of the robot is below pain and injury thresholds $T_{\text{safe}}(t)$, resulting in the PFL safety constraint

$$c_{\text{safe,PFL}}(t) := \overline{c_{\text{coll}}(t)} \vee (T_{\text{robot}}(t) \leq T_{\text{safe}}(t)). \quad (3)$$

The safe energy threshold $T_{\text{safe}}(t)$ can depend on the collision type (constrained collisions for clamping or unconstrained collisions for free movement), the shape of the robot part in contact (e.g., sharp or blunt edges), or the human body part affected in the worst case [31], [47], [48]. In contrast, speed and separation monitoring (SSM) does not permit any contact with humans, targeting coexistence without close interaction.

If a collision can occur, the robot must be at a complete stop with zero kinetic energy, i.e., $T_{\text{safe}}(t) = 0$ in (3).

V. PATH-CONSISTENT SAFETY FILTER

We now detail PACS, which ensures the safe deployment of pretrained DPs. An overview is provided in Figure 2.

A. Safety-Permitting Diffusion Policy

To maintain high task performance despite path-consistent safety interventions, we need to design the policy in such a way that slowing down or stopping the robot does not in itself represent an out-of-distribution situation. To achieve this, we only include RGB images $\mathbf{I}(t)$ and joint angles $\mathbf{q}(t)$ into the observation $\mathbf{o}(t)$ and no explicit velocity information, which is a common approach in recent works [6], [7]. Our approach is applicable to various action spaces, but we restrict our presentation to joint velocity control for clarity. In this work, the actions are the delta joint positions $\mathbf{a}_i = \Delta \hat{\mathbf{q}}_i = (\hat{\mathbf{q}}_{i+1} - \hat{\mathbf{q}}_i)$ and we use $\hat{\cdot}$ to distinguish the desired actions generated by the policy from the actual safe commands sent to the robot. Without loss of generality, we reset the clock in each step of the planning algorithm and safety filter, i.e., $t_0 = 0$. By integrating the desired joint positions forward in time, starting from the current joint position $\mathbf{q}(0)$, we compute a sequence of waypoints

$$\hat{\mathbf{p}}(0) = (\hat{\mathbf{q}}_0, \dots, \hat{\mathbf{q}}_h), \quad \text{where } \hat{\mathbf{q}}_0 = \mathbf{q}(0), \quad (4)$$

describing the desired path (Algorithm 1, line 6).

B. Path-Consistent Safety for Diffusion Policies

To ensure that our safety filter does not steer the policy into out-of-distribution states, we adopt a path-consistent safety formulation. The key idea is to guarantee safety by modifying the joint speed $\dot{\mathbf{q}}(t)$ and acceleration $\ddot{\mathbf{q}}(t)$ while

Algorithm 1 Path-consistent safety filter for diffusion policies (DPs) and vision-language-action models (VLAs).

Require: DP or VLA $\pi(\cdot)$, chunk length H , safety filter horizon h , policy timestep Δt , safety step size α_s , initial configuration $\mathbf{q}(0) \in \mathcal{X}_{\text{Safe}}$

```

1:  $t \leftarrow 0$ 
2:  $\text{success} \leftarrow \text{False}$ 
3: while  $\neg \text{success}$  ▷ Policy steps
4:    $\mathbf{o}(t) \leftarrow \text{GETOBSERVATION}(t)$ 
5:    $\mathbf{A}(t) = (\Delta \hat{\mathbf{q}}_0, \dots, \Delta \hat{\mathbf{q}}_{H-1}) \sim \pi(\cdot | \mathbf{o}(t))$ 
6:    $(\hat{\mathbf{q}}_0, \dots, \hat{\mathbf{q}}_h) \leftarrow \text{INTEGRATE}(\mathbf{A}(t))$ 
7:    $\chi_I([t_0, t_h]) \leftarrow \text{COMPUTETRAJECTORY}((\hat{\mathbf{q}}_0, \dots, \hat{\mathbf{q}}_h))$ 
8:    $\text{shield} \leftarrow \text{SETTRAJECTORY}(\chi_I([t_0, t_h]))$ 
9:   for  $k \leftarrow 0$  to  $\frac{h\Delta t}{\alpha_s}$  do ▷ Safety steps
10:     $\mathbf{m}_o(t) \leftarrow \text{GETOBJECTMEASUREMENTS}(t)$ 
11:     $\mathbf{u}_{\text{safe}}(t + \alpha_s) \leftarrow \text{SHIELDSTEP}(t, \mathbf{m}_o(t))$ 
12:     $\text{APPLYACTION}(\mathbf{u}_{\text{safe}}(t + \alpha_s))$ 
13:     $t \leftarrow t + \alpha_s$ 
14:   end for
15:    $\text{success} \leftarrow \text{CHECKSUCCESS}(\mathbf{o}(t))$ 
16: end while

```

staying on the intended trajectory. Existing path-consistent safety filters for learning-based policies using reachability analysis [30], [31] can only take as action input a single goal position with zero velocity. However, DPs and vision-language-action models predict a chunk $\mathbf{A}(t)$ of multiple consecutive actions together and operate at a relatively low update frequency $1/(h\Delta t)$. Therefore, directly applying single-action approaches [30], [31] to these policies either leads to a very slow execution by stopping at each waypoint or leaves out intermediate action steps, sacrificing fine-grained control and causing a deviation from the desired behavior.

To construct smooth trajectories that align with the predicted action chunk, we propose an intended trajectory planning module (cf. Figure 2). For each action chunk $\mathbf{A}(t)$ starting from the initial time step $t_0 = 0$, we compute a time-optimized intended trajectory $\chi_I([0, t_h]) = \{\mathbf{x}(\tau) | \tau \in [0, t_h]\}$ (Algorithm 1, line 7), where $\mathbf{x}(\tau) = (\mathbf{q}(\tau), \dot{\mathbf{q}}(\tau), \ddot{\mathbf{q}}(\tau), \ddot{\ddot{\mathbf{q}}}(\tau))$ is the state of the system, by solving the optimization problem

$$\min_{t_h, \chi_I([0, t_h])} t_h \quad (5a)$$

$$\text{s.t.} \quad \forall \tau \in [0, t_h] : \mathbf{x}(\tau) \in \mathcal{X}, \quad (5b)$$

$$\forall k \in \{0, \dots, h\} : \mathbf{q}(t_k) = \hat{\mathbf{q}}_k, \quad (5c)$$

where the set \mathcal{X} defines robot-specific kinematic and dynamic constraints $\mathbf{q}_{\min} \leq \mathbf{q}(\tau) \leq \mathbf{q}_{\max}$, $\|\dot{\mathbf{q}}(\tau)\| \leq \dot{q}_{\max}$, $\|\ddot{\mathbf{q}}(\tau)\| \leq \ddot{q}_{\max}$, and $\|\ddot{\ddot{\mathbf{q}}}(\tau)\| \leq \ddot{\ddot{q}}_{\max}$. Adjusting the velocity, acceleration, and jerk limits allows us to adapt the execution speed of the action chunk, which gives us a more fine-grained control over the execution speed of the policy compared to modifying the actions directly. Importantly, our intermediate planning module provides a trajectory by

TABLE I
ROBOMIMIC TASK SUCCESS RATE RESULTS ON 100 ROLLOUTS.

Method	Safe?	LIFT	CAN	SQUARE	Average
Operational space contr. OFF	×	1.00	0.99	0.74	0.91
Contr. barrier func. [13]	✓	0.11	0.00	0.00	0.04
Single-action - SSM	✓	0.97	0.26	0.00	0.41
Single-action - PFL	✓	0.94	0.33	0.04	0.44
PACS (ours) - SSM	✓	0.97	0.80	0.30	0.69
PACS (ours) - PFL	✓	0.93	0.85	0.38	0.72

solving (5), which we forward to our path-consistent safety filter (Algorithm 1, line 8).

The safety filter (Algorithm 1, line 9-14) operates at a high frequency and formally guarantees constraint satisfaction while staying on the desired path (4). Based on the current state of the robot $\mathbf{x}_0 = \mathbf{x}(0)$ and the measured object positions $\mathbf{m}_o(t)$ (Algorithm 1, line 10), we compute a path-consistent monitored trajectory [31] by concatenating an intended trajectory (desired or recovery motion) $[0, t_I]$ and a failsafe trajectory (stopping motion) $[t_I, t_F]$

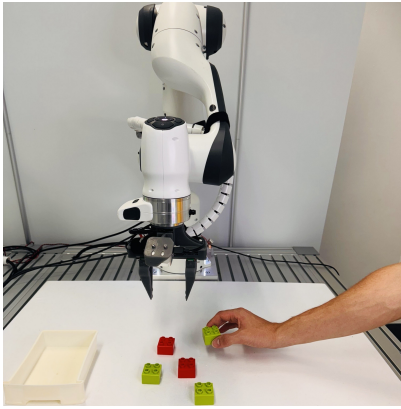
$$\chi_M(t) = \begin{cases} \chi_I(\tau; \mathbf{x}_0, \mathbf{u}_I(\cdot)), & \tau \in [0, t_I], \\ \chi_F(\tau; \mathbf{x}_I, \mathbf{u}_F(\cdot)), & \tau \in [t_I, t_F]. \end{cases} \quad (6)$$

At each safety timestep of duration α_s , we verify the monitored trajectory (6) for the task-specific safety constraints as described in Section IV-B. If the verification is successful, we execute the intended trajectory $\mathbf{u}_{\text{safe}}(t) = \mathbf{u}_I(t)$; otherwise, we proceed with the last successfully verified failsafe trajectory that brings the system to an invariable safe set, i.e., $\mathbf{u}_{\text{safe}}(t) = \mathbf{u}_F(t)$. We repeat this process until receiving a new action chunk from the policy after $h\Delta t/\alpha_s$ safety steps. Since this safeguarding mechanism is provably safe by induction, as shown in prior work [31], PACS enables the safe deployment of DPs.

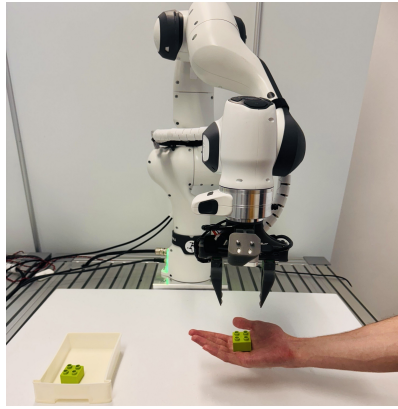
VI. EVALUATION

We conduct extensive experiments in simulation and on hardware to evaluate the real-time safeguarding capabilities of our method. Our code will be released upon acceptance to preserve anonymity. Our quantitative evaluation primarily aims to validate the following hypotheses:

- H1** Through behavior cloning from demonstrations alone, DPs cannot learn a safe behavior for interaction with moving objects.
- H2** With our path-consistent safeguarding method, the task success rate remains close to the unsafe policy ($\pm 5\%$) in our conducted experiments.
- H3** Computing intended trajectories from entire action chunks leads to higher task success rates than single-action approaches.
- H4** PACS is real-time capable and does not reduce task execution speed, unless required to ensure safety.



(a) SORTING: Put the red blocks into the box while a person is taking the green blocks.



(b) HANDOVER: Grab blocks from a moving hand and put them into the box.



(c) FEEDING: Carefully put a fork with food in the mouth and pull the fork back out.

Fig. 3. Visualization of our three real-world tasks, which require safe and reactive motion in close proximity to the human body. SORTING represents a coexistence task, where no collisions are permitted, whereas HANDOVER and FEEDING are collaboration tasks requiring non-harmful, low-force interactions.

A. Simulation Experiments

First, we evaluate the performance of PACS against an unsafe operational space controller and a control barrier function baseline [13] on the three robomimic [49] benchmark tasks LIFT, CAN, and SQUARE. We integrated these tasks in human-robot-gym [50] and added a spherical dynamic object that moves in pre-defined patterns on the surface of the table. The goal is to fulfill the original task while ensuring the safety constraints $c_{\text{safe,SSM}}$ (for operational space controller, control barrier function, and SSM) or $c_{\text{safe,PFL}}$ (for PFL). To evaluate **H3**, we further tested a single-action version of PACS that sequentially computes new intended trajectories from each action in the action chunk.

We trained the DP with a temporal UNet architecture [3] on the provided dataset of 200 proficient human teleoperated demonstrations per task. The policy receives the end-effector and object poses of the last two timesteps as observations and predicts desired changes in the end effector pose and gripper opening as actions. The operational space controller directly controls the robot in Cartesian space to follow these goals. For all other approaches, we used an inverse kinematics solver [51] to translate the desired changes in end-effector pose into desired joint state actions and control the robot in joint space. Additional implementation details can be found in Table III. For the operational space controller baseline, we used $h = 8$ action execution steps and applied each action for $\Delta t = 50$ ms. To evaluate the impact of the safety mechanism and the inverse kinematics translation separately, we also tested the performance of our safety shield in OFF mode. Here, we compute the same intended trajectory in joint space, but never execute a failsafe trajectory.

The results in Table I highlight that PACS achieves a significantly higher task success rate than the reactive control barrier function safety filter. Comparing the safe SSM and PFL mode with the controller in the OFF setting, we see that path-consistent braking has no significant impact on the performance of the policy, **confirming H2**. However,

TABLE II
CONSTRAINTS OF OUR REAL-WORLD TASKS.

Task	Constraint Type	Body Part	Condition	T_{safe} [J]
SORTING	SSM	Hand	-	0.000
HANDOVER	PFL	Hand	Constrained	0.014
			Unconstrained	0.265
FEEDING	PFL	Head (Eye)	Unconstrained	0.001

if we compare OFF to the operational space controller, we see a noticeable drop in performance in the SQUARE task, which is most likely due to slight inaccuracies induced by transforming the end-effector poses into joint space. Table I further shows that the action chunk trajectories of PACS improve the performance of the sequentially executed single actions by 28%, **confirming H3**.

B. Real-World Setup

Second, we conducted hardware experiments with a Franka FR3 manipulator for HRI, as safety is particularly critical in this area.

1) *Tasks*: We considered three tasks requiring safe motion in close proximity to humans, as visualized in Figure 3:

- SORTING: We placed two red and three green blocks randomly on the table. The robot must put the red blocks into a box while the human is taking the green blocks. We investigated different human motion patterns and speeds. To ensure safety, no collisions between the robot and the human are allowed ($c_{\text{safe,SSM}}$).
- HANDOVER: The robot must pick a block from the hand of a human partner. To ensure safety, the energy of the impact between the gripper and the hand must not exceed a given energy limit ($T_{\text{safe,hand}}$).
- FEEDING: The robot must put a fork with food into the mouth of the user. To ensure safety, the energy of the impact between the head and the fork must not exceed a given limit ($T_{\text{safe,head}}$). Note that $T_{\text{safe,head}} \ll T_{\text{safe,hand}}$.

TABLE III
IMPLEMENTATION DETAILS FOR THE DP AND
VISION-LANGUAGE-ACTION MODEL IN OUR EXPERIMENTS.

Hyperparameter	DP (sim)	DP (real)	SmolVLA
Demonstrat. per task	200	50	50
Generative model	Diffusion	Diffusion	Flow Matching
Architecture	U-Net	U-Net	VLM + Transformer
Training steps	64k	20k	40k
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch size	100	64	64
Action chunk len. H	16	16	50
Action exec. steps h	2	6	6
Inference timesteps	100 (DDPM)	10 (DDIM)	10 (Euler)
Sampling time Δt	100 ms	33 ms	33 ms

We use a printed AI-generated face with a cut-out mouth for our quantitative experiments and conduct qualitative tests with a real human.

2) *Safety Constraints*: We used the SaRA framework [52] to compute reachable occupancies and performed real-time intersection checking. The safe energy thresholds in (3) have been formally identified for the different tasks [31]. Further details and numerical values are provided in Table II. We leveraged Ruckig [53] to compute the intended trajectory (5), using the standard joint limits of the FR3, velocity limits between 1.25 and 2 rad s^{-1} , an acceleration limit of 10 rad s^{-2} and a jerk limit of 400 rad s^{-3} for each joint.

3) *Policies*: We deployed two generative policies: DP [3] using denoising diffusion [1] and SmolVLA [7] using flow matching [2]. The observations consisted of joint angles and the images of a wrist and a third-person camera, and the actions were desired changes in joint angles and the gripper opening. We provide details on the training data and the policy implementation in Table III. CRISP [54] has been used for data recording and for translating the joint commands to motor torques. The SmolVLA policy struggled with SORTING and FEEDING, likely due to the high precision required; therefore, we only evaluated it on the HANDOVER task.

4) *Metrics*: We report the success rate of completing the task (Success) and the success rate of completing the task while remaining safe at all times (Safe Success) based on 30 test rollouts. Moreover, we report the proportion of timesteps for which the task-specific safety constraints are violated (Safety Viol.) and the duration of successful rollouts.

C. Real-World Results

We deployed the policies with our safety filter OFF and active, similar to the simulation experiments. As shown by the results in Table IV, the policies achieved very similar task success rates of about 80% on average with and without our safety filter, **supporting H2**. The safeguarded policies do not exhibit constraint violations in our experiments. In contrast, the nominal policies are in an unsafe state 56% of the time with each rollout containing constraint violations, resulting in a Safe Success of 0%, which **confirms H1**. These results

TABLE IV
IMPACT OF SAFEGUARDING IN OUR REAL-WORLD EXPERIMENTS.

Task	Policy	Safety Filter	Success	Safe Succ.	Safety Viol.
SORTING	DP	OFF	0.77	0.00	0.67 ± 0.12
		PACS (ours)	0.80	0.80	0.00 ± 0.00
HANDOVER	DP	OFF	1.00	0.00	0.32 ± 0.07
		PACS (ours)	0.97	0.97	0.00 ± 0.00
	SmolVLA	OFF	0.77	0.00	0.41 ± 0.15
		PACS (ours)	0.80	0.80	0.00 ± 0.00
FEEDING	DP	OFF	0.63	0.00	0.85 ± 0.04
		PACS (ours)	0.63	0.63	0.00 ± 0.00
Average		OFF	0.79	0.00	0.56 ± 0.21
		PACS (ours)	0.80	0.80	0.00 ± 0.00

highlight the necessity of online safety mechanisms when deploying DPs in dynamic environments and the effectiveness of PACS for this purpose.

We also compare our path-consistent safety approach with a popular reactive method: control barrier functions [13]. Since control barrier functions are better suited for coexistence than for collaboration tasks, we conduct this comparison for the SORTING task. As can be seen from Table V, PACS achieved a 37% higher success rate than the control barrier function-based filter. To further examine this inferior performance, we visualize the training data distribution and the end-effector positions of randomly selected rollouts in Figure 4. While our method kept the policy in distribution, the control barrier function often pushed the robot into out-of-distribution states from which the policy struggles to recover, leading to a higher failure rate. Based on our simulation results, we find that PACS overall achieved a 51% higher task success rate than the safety filter using a control barrier function. Besides, we observe that policy recovery usually takes a few seconds, leading to a higher average task completion time. The average computation time of one safety step with PACS was 0.20 ms compared to 0.64 ms for the control barrier function, and recalculating the intended trajectory took about 5 ms. This **supports H4** and shows that our method can be deployed in real time in highly dynamic environments.

Next, we evaluate the impact of our proposed intermediate trajectory generation (Algorithm 1, line 7) on task execution speed. We conducted this ablation study without a person in the workspace, as the safety filter could otherwise slow down the robot. To this end, we defined a modified version of the SORTING task, in which the green blocks are not picked up by a person. Moreover, to eliminate the effect of the initial block placement on execution time, we always used the same starting configuration. For comparison, we used the standard DP setup in which the generated actions are directly sent to the low-level controller at a frequency of 30 Hz instead of performing a time parameterization via (5). The results of this experiment are provided in Table V. Both approaches achieved the same success rate of 93%, which is higher than in Table IV; likely due to the absence of the human and

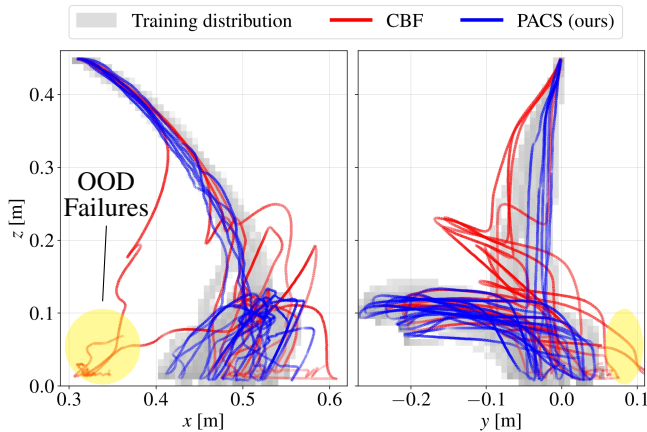


Fig. 4. End-effector paths for the SORTING task. The color intensity of the trajectories indicates the velocity, and the shaded grey areas visualize the training distribution. Our safety filter reduces the velocity without leaving the desired path when the human is nearby. In contrast, the control barrier function (CBF) pushes the robot away from unsafe states, which often leads to out-of-distribution (OOD) states from which the policy cannot recover.

TABLE V

BASELINE COMPARISON AND SPEED ANALYSIS FOR THE SORTING TASK.

Human	Safety Filter	Success	Duration [s]
w/ human	Control barrier function	0.43	35.8 ± 18.8
	PACS (ours)	0.80	32.6 ± 11.1
w/o human	-	0.93	25.2 ± 3.3
	PACS (ours)	0.93	21.7 ± 3.9

the fixed initial block configuration. Yet, the intermediate trajectory generation reduced the average task execution time by 14% from 25.2s to 21.7s and increased the average Cartesian end-effector speed by 13%. This **confirms H4** and even shows that ensuring the kinematic and dynamic feasibility of the generated plan can lead to faster task execution without sacrificing performance.

Lastly, we tested our safety filter for the FEEDING task with a real human. As shown in our accompanying video, the robot slowed down when approaching the human, and the safeguarded policy inserted and withdrew the fork carefully and precisely. These experiments highlight the potential of our approach to harness the expressiveness of DPs for highly safety-critical applications, such as healthcare.

VII. CONCLUSION

We propose PACS, a general framework for safely deploying imitation learning policies using action chunks, such as DPs and vision-language-action models, in dynamic environments. Our insight is that safety interventions should be path-consistent to avoid out-of-distribution situations and maintain high task performance. We achieve this by computing an intended trajectory from the action chunk and monitoring it using reachability analysis. Our experiments show that PACS can guarantee safety in real time and achieve high task success even on challenging real-world HRI tasks, whereas reactive safety filtering with control barrier functions frequently

leads the policy into unrecoverable states. Our method is specifically designed to satisfy dynamic constraints, such as moving obstacles. Handling (semi-)static obstacles via constraint-aware online replanning is an interesting avenue for future work. We consider this work to be an important step towards deploying generative policies in safety-critical, human-centered environments.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge financial support of the research training group ConVeY funded by the German Research Foundation under grant GRK 2428; financial support by the DAAD program Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Research, Technology and Space (BMFTR); and support by the Robotics Institute Germany, funded by BMFTR grant 16ME0997K.

REFERENCES

- [1] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.
- [2] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2023.
- [3] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Proc. of Robotics: Science and Systems (RSS)*, 2023.
- [4] M. Reuss, Ö. E. Yagmurlu, F. Wenzel, and R. Lioutikov, “Multimodal diffusion transformer: Learning versatile behavior from multimodal goals,” in *Proc. of Robotics: Science and Systems (RSS)*, 2024.
- [5] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, “Open X-embodiment: Robotic learning datasets and RT-X models,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 6892–6903.
- [6] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, “ $\pi 0$: A vision-language-action flow model for general robot control,” in *Proc. of Robotics: Science and Systems (RSS)*, 2025.
- [7] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti *et al.*, “SmolVLA: A vision-language-action model for affordable and efficient robotics,” *arXiv preprint arXiv:2506.01844*, 2025.
- [8] X. Ye, R. H. Yang, J. Jin, Y. Li, and A. Rasouli, “RA-DP: Rapid adaptive diffusion policy for training-free high-frequency robotics replanning,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2025, pp. 6227–6234.
- [9] Z. Wang, M. Li, A. Mandlekar, Z. Xu, J. Fan, Y. Narang, L. Fan, Y. Zhu, Y. Balaji, M. Zhou *et al.*, “One-step diffusion policy: Fast visuomotor policies via diffusion distillation,” in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2025.
- [10] 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, no. 1, pp. 411–444, 2022.
- [11] H. Krasowski, J. Thumm, M. Müller, L. Schäfer, X. Wang, and M. Althoff, “Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking,” *Transactions on Machine Learning Research*, 2023.
- [12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proc. of the European Control Conf. (ECC)*, 2019, pp. 3420–3431.
- [13] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, “Safety-critical manipulation for collision-free food preparation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10954–10961, 2022.
- [14] K. P. Wabersich and M. N. Zeilinger, “A predictive safety filter for learning-based control of constrained nonlinear dynamical systems,” *Automatica*, vol. 129, 2021.

- [15] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, “A predictive safety filter for learning-based racing control,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7635–7642, 2021.
- [16] C. Agia, R. Sinha, J. Yang, Z. Cao, R. Antonova, M. Pavone, and J. Bohg, “Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress,” in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025, pp. 689–723.
- [17] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. Ambrus, H. Nishimura, and M. Itkina, “Can we detect failures without failure data? Uncertainty-aware runtime failure detection for imitation learning policies,” in *Proc. of Robotics: Science and Systems (RSS)*, 2025.
- [18] R. Römer, A. Kobras, L. Worbis, and A. P. Schoellig, “Failure prediction at runtime for generative robot policies,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
- [19] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira, “Effortless creation of safe robots from modules through self-programming and self-verification,” *Science Robotics*, vol. 4, no. 31, 2019.
- [20] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, “Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems,” *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.
- [21] K.-C. Hsu, H. Hu, and J. F. Fisac, “The safety filter: A unified view of safety-critical control in autonomous systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, pp. 47–72, 2024.
- [22] S. Gros, M. Zanon, and A. Bemporad, “Safe reinforcement learning via projection on a safe set: How to achieve optimality?” in *Proc. of the IFAC World Congress*, 2020, pp. 8076–8081.
- [23] Z. Marvi and B. Kiumarsi, “Safe reinforcement learning: A control barrier function optimization approach,” *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.
- [24] D. Morton and M. Pavone, “Safe, task-consistent manipulation with operational space control barrier functions,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2025, pp. 187–194.
- [25] M. Althoff, “An introduction to CORA 2015,” in *EPiC Series in Computing*, vol. 34, 2015, pp. 120–151.
- [26] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, “Reachability-based safe learning with Gaussian processes,” in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2014, pp. 1424–1431.
- [27] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.
- [28] H. Krasowski, X. Wang, and M. Althoff, “Safe reinforcement learning for autonomous lane changing using set-based prediction,” in *Proc. of the IEEE Int. Intelligent Transportation Systems Conf. (ITSC)*, 2020, pp. 1–7.
- [29] Y. S. Shao, C. Chen, S. Kousik, and R. Vasudevan, “Reachability-based trajectory safeguard (RTS): A safe and fast reinforcement learning safety layer for continuous control,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3663–3670, 2021.
- [30] J. Thumm and M. Althoff, “Provably safe deep reinforcement learning for robotic manipulation in human environments,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 6344–6350.
- [31] J. Thumm, J. Balletshofer, L. Maglanoc, L. Muschal, and M. Althoff, “A general safety framework for autonomous manipulation in human environments,” *IEEE Transactions on Robotics*, vol. Accepted for Publication, 2025.
- [32] A. M. Zanchettin and P. Rocco, “Path-consistent safety in mixed human-robot collaborative manufacturing environments,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013, pp. 1131–1136.
- [33] M. J. Zeestraten, A. Pereira, M. Althoff, and S. Calinon, “Online motion synthesis with minimal intervention control and formal safety guarantees,” in *Prof. of the IEEE International Conf. on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 2116–2121.
- [34] L. Brunke, Y. Zhang, R. Römer, J. Naimer, N. Staykov, S. Zhou, and A. P. Schoellig, “Semantically safe robot manipulation: From semantic scene understanding to motion safeguards,” *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4810–4817, 2025.
- [35] W. Jung, D. Anthony, U. A. Mishra, N. R. Arachchige, M. Bronars, D. Xu, and S. Kousik, “RAIL: Reachability-aided imitation learning for safe policy execution,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 3582–3589.
- [36] J. Carvalho, A. T. Le, P. Kicki, D. Koert, and J. Peters, “Motion planning diffusion: Learning and adapting robot motion planning with diffusion models,” *IEEE Transactions on Robotics*, vol. 41, pp. 4881–4901, 2025.
- [37] M. Nikken, N. Botteghi, W. Roozing, and F. Califano, “Denoising diffusion planner: Learning complex paths from low-quality demonstrations,” *arXiv preprint arXiv:2410.21497*, 2024.
- [38] R. Römer, L. Brunke, M. Schuck, and A. P. Schoellig, “Safe offline reinforcement learning using trajectory-level diffusion models,” in *ICRA 2024 Workshop Back to the Future: Robot Learning Going Probabilistic*, 2024.
- [39] R. Römer, A. v. Rohr, and A. P. Schoellig, “Diffusion predictive control with constraints,” in *Proc. of the Ann. Learning for Dynamics and Control Conf. (LADC)*, vol. 283, 2025, pp. 791–803.
- [40] J. Liang, J. K. Christopher, S. Koenig, and F. Fioretto, “Simultaneous multi-robot motion planning with projected diffusion models,” in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2025.
- [41] W. Xiao, T.-H. Wang, C. Gan, R. Hasani, M. Lechner, and D. Rus, “SafeDiffuser: Safe planning with diffusion probabilistic models,” in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2023.
- [42] T.-Y. Huang, A. Lederer, D.-J. Wu, X. Dai, S. Zhang, S. Sosnowski, S.-H. Sun, and S. Hirche, “SAD-Flower: Flow matching for safe, admissible, and dynamically consistent planning,” *arXiv preprint arXiv:2511.05355*, 2025.
- [43] T. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” in *Proc. of Robotics: Science and Systems (RSS)*, 2023.
- [44] S. B. Liu, B. Schürmann, and M. Althoff, “Guarantees for real robotic systems: Unifying formal controller synthesis and reachset-conformant identification,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3776–3790, 2023.
- [45] S. B. Liu and M. Althoff, “Online verification of impact-force-limiting control for physical human-robot interaction,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 777–783.
- [46] D. Beckert, A. Pereira, and M. Althoff, “Online verification of multiple safety criteria for a robot trajectory,” in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2017, pp. 6454–6461.
- [47] R. J. Kirschner, C. M. Micheler, Y. Zhou, S. Siegner, M. Hamad, C. Glowalla, J. Neumann, N. Rajaei, R. Burgkart, and S. Haddadin, “Towards safe robot use with edged or pointed objects: A surrogate study assembling a human hand injury protection database,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 12 680–12 687.
- [48] R. J. Kirschner, J. Yang, E. Elshani, C. M. Micheler, T. Leibbrand, D. Müller, C. Glowalla, N. Rajaei, R. Burgkart, and S. Haddadin, “Towards unconstrained collision injury protection data sets: Initial surrogate experiments for the human hand,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2024, pp. 14 012–14 019.
- [49] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” in *Proc. of the Conf. on Robot Learning (CoRL)*, 2021, pp. 1678–1690.
- [50] J. Thumm, F. Trost, and M. Althoff, “Human-robot gym: Benchmarking reinforcement learning in human-robot collaboration,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 7405–7411.
- [51] E. Coumans and Y. Bai, “Pybullet, a Python module for physics simulation for games, robotics and machine learning,” 2016–2021. [Online]. Available: <http://pybullet.org>
- [52] S. R. Schepp, J. Thumm, S. B. Liu, and M. Althoff, “SaRA: A tool for safe human-robot coexistence and collaboration through reachability analysis,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 4312–4317.
- [53] L. Berscheid and T. Kröger, “Jerk-limited real-time trajectory generation with arbitrary target states,” in *Proc. of Robotics: Science and Systems (RSS)*, 2021.
- [54] D. S. J. Pro, O. Hausdörfer, R. Römer, M. Dösch, M. Schuck, and A. P. Schoellig, “CRISP – compliant ROS2 controllers for learning-based manipulation policies and teleoperation,” *arXiv preprint arXiv:2509.06819*, 2025.