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

Ralf Römer*,[1], Julian Balletshofer*,[1], Jakob Thumm[2], Marco Pavone[2], Angela P. Schoellig[1], and Matthias Althoff[1]

***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

Recently, imitation learning using diffusion models [1] and flow matching [2] has enabled robots to successfully perform 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 (CBFs) [12], [13] or predictive safety filters [14]. Most of these are reactive mechanisms that satisfy safety constraints by changing the robot's path. However, as illustrated in Figure 1, this often drives the system into unseen states. DPs are particularly prone to errors in these out-of-distribution (OOD) cases, increasing the risk of task failure [15]–[17]. To avoid this OOD situation, safety mechanism interventions should ideally keep the system close to the policy's desired actions.

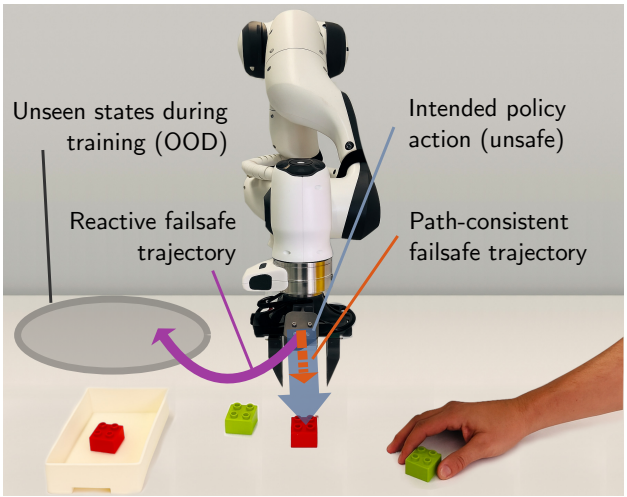


Fig. 1. Deploying DPs in dynamic environments with moving objects requires safeguarding mechanisms. Reactive strategies, such as control barrier functions, often drive the robot into out-of-distribution (OOD) states not seen during training, leading to unpredictable behavior. We propose that safety mechanisms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

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. 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 [18] 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 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 leads to a 28 % improvement in task success rates compared to sequentially treating each action individually.

*Equal contribution.

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

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

II. Related Work

Safety filters [10], [11], [19] ensure safe deployment by verifying whether a failsafe action exists after the proposed action. Model predictive safety filters [14] and CBFs [12], [13] formulate optimization problems to find the safest input, while reachability-based methods [20], [21] provably maintain a failsafe stopping trajectory. Regarding safety of DPs, RAIL [22] maintains a reachability-based failsafe but distorts the action distribution and only handles static constraints. Methods injecting safety into the denoising process [23]–[25] provide some guarantees but incur high computational costs, limiting applicability to high-dimensional or real-time systems. 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 $\boldsymbol{x}(t) \in \mathcal{X}$ and input $\boldsymbol{u}(t) \in \mathcal{U}$ that should perform a given task while adhering to safety constraints. We assume that the state $\boldsymbol{x}$ 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 expert demonstrations $\mathcal{D} = \{\boldsymbol{o}_k, \boldsymbol{a}_k\}_{k=0}^{K}$ in the form of observation-action pairs, where the observations $\boldsymbol{o} \in \mathcal{O}$ may include robot joint angles, camera images, language commands, and other environmental information. From these demonstrations, a DP $\pi : \mathcal{A}^H \times \mathcal{O} \to \mathbb{R}_+$ is trained that returns a distribution over action chunks $\boldsymbol{A} \in \mathcal{A}^H$ of length H. We aim to design a controller $\boldsymbol{k} : \mathcal{X} \times \mathcal{A}^H \times \mathbb{R} \to \mathcal{U}$ that takes the latest action chunk $\boldsymbol{A}$ as input and returns a safe control input $\boldsymbol{u}_{\text{safe}}(t)$, while minimizing deviation from the intended policy behavior.

IV. Preliminaries

A. Diffusion Policies with Action Chunking

A DP π is trained to approximate the conditional action distribution in expert demonstrations $\mathcal{D}$ [3]. New actions are generated by iteratively transforming random noise into the learned target distribution [1], [2]. Most DPs return a chunk of H consecutive actions $\boldsymbol{A}_k = (\boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \ldots, \boldsymbol{a}_{k+H-1})$ to improve temporal consistency and execution speed [3], [26]. During deployment, we iteratively sample new action chunks $\boldsymbol{A}_k \sim \pi(\cdot \mid \boldsymbol{o}_k)$, apply the first $h \leq H$ actions, and replan after time $h\Delta t$. This formulation encompasses single-task DPs [3], [4] and generalist vision-language-action models [6], [7].

B. Safety Constraints for Dynamic Objects

We define safety using set-based reachability analysis [18]. The *reachable occupancy* $\mathcal{C}(t)$ captures all possible spatial extents of a system at time t. Following prior work [18], a collision event is

$$c_{\text{coll}}(t) \coloneqq \mathcal{C}_{\text{object}}(t) \cap \mathcal{C}_{\text{robot}}(t) \neq \emptyset. \tag{1}$$

We define two safety modes in accordance with ISO/TS 15066:2016. Power and force limiting (PFL) allows contacts with humans as long as the robot's kinetic energy remains below a pain threshold $T_{\text{safe}}(t)$:

$$c_{\text{safe,PFL}}(t) \coloneqq \overline{c_{\text{coll}}(t)} \vee \big(T_{\text{robot}}(t) \leq T_{\text{safe}}(t)\big). \tag{2}$$

Speed and separation monitoring (SSM) does not permit any contact ($T_{\text{safe}}(t) = 0$), targeting coexistence scenarios.

V. Path-Consistent Safety Filter

A. Safety-Permitting Diffusion Policy

To maintain high task performance despite path-consistent safety interventions, we need to design the policy such that slowing down or stopping the robot does not itself represent an OOD situation. To achieve this, we include only RGB images $\boldsymbol{I}(t)$ and joint angles $\boldsymbol{q}(t)$ in the observation $\boldsymbol{o}(t)$, with no explicit velocity information [6], [7]. This is a common approach in recent VLA works [6], [7] and ensures the policy does not notice when the robot is slowed down. The actions are delta joint positions $\boldsymbol{a}_i = \Delta\hat{\boldsymbol{q}}_i = (\hat{\boldsymbol{q}}_{i+1} - \hat{\boldsymbol{q}}_i)$, and integrating the action chunk from the current joint position $\boldsymbol{q}(0)$ yields a sequence of waypoints

$$\hat{\boldsymbol{p}}(0) = (\hat{\boldsymbol{q}}_0, \ldots, \hat{\boldsymbol{q}}_h), \quad \text{where} \quad \hat{\boldsymbol{q}}_0 = \boldsymbol{q}(0), \tag{3}$$

describing the desired path along which PACS will modulate the robot's speed.

B. Path-Consistent Safety for Diffusion Policies

An overview is provided in Figure 2 and Algorithm 1. The key idea is to guarantee safety by modifying joint speed $\dot{\boldsymbol{q}}(t)$ and acceleration $\ddot{\boldsymbol{q}}(t)$ while staying on the intended trajectory. Existing path-consistent safety filters [20], [21] can only take a single goal position as input; directly applying them to DPs leads to either very slow execution by stopping at each waypoint, or loss of fine-grained control.

To construct smooth trajectories aligned with the predicted action chunk, we propose an intended trajectory planning module. For each action chunk starting from $t_0 = 0$, we compute a time-optimized intended trajectory $\chi_I([0, t_h])$ by solving

$$\min_{t_h,\, \chi_I([0,t_h])} \quad t_h \tag{4a}$$

$$\text{s.t.} \quad \forall \tau \in [0, t_h] : \ \boldsymbol{x}(\tau) \in \mathcal{X}, \tag{4b}$$

$$\forall k \in \{0, \ldots, h\} : \boldsymbol{q}(t_k) = \hat{\boldsymbol{q}}_k, \tag{4c}$$

where $\mathcal{X}$ defines robot-specific kinematic and dynamic constraints on joint positions, velocities, accelerations, and jerks. Adjusting these limits allows fine-grained control over execution speed.

The safety filter (Algorithm 1, lines 7–10) operates at high frequency and formally guarantees constraint satisfaction while staying on the desired path. Based on the current robot state $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ and measured object positions $\boldsymbol{m}_{\text{o}}(t)$, it constructs a path-consistent monitored trajectory by concatenating an intended trajectory $[0, t_{\text{I}}]$ and a failsafe stopping

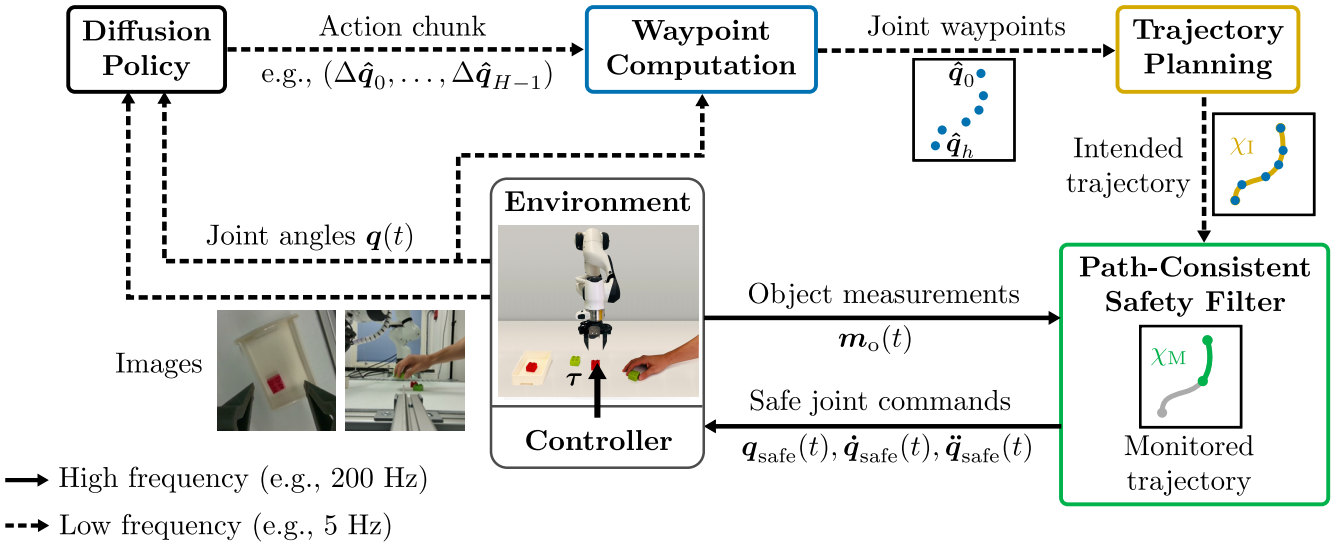


Fig. 2. System overview of PACS. The policy generates action chunks that are transformed into a sequence of desired waypoints. From these waypoints, we compute a kinematically and dynamically feasible intended trajectory. PACS continuously monitors this trajectory and applies high-frequency safety filtering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

Algorithm 1 Path-consistent safety filter for DPs and VLAs.

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

$t \leftarrow 0$; *success* $\leftarrow$ **False**
while $\neg$*success* **do** ▷ Policy steps
  $\boldsymbol{A}(t) = (\Delta\hat{\boldsymbol{q}}_0, \ldots, \Delta\hat{\boldsymbol{q}}_{H-1}) \sim \pi(\cdot \mid \boldsymbol{o}(t))$
  $(\hat{\boldsymbol{q}}_0, \ldots, \hat{\boldsymbol{q}}_h) \leftarrow \textsc{Integrate}(\boldsymbol{A}(t))$
  $\chi_I([t_0, t_h]) \leftarrow \textsc{ComputeTrajectory}((\hat{\boldsymbol{q}}_0, \ldots, \hat{\boldsymbol{q}}_h))$
  shield $\leftarrow \textsc{SetTrajectory}(\chi_I([t_0, t_h]))$
  for $k \leftarrow 0$ **to** $\frac{h\Delta t}{\alpha_s}$ **do** ▷ Safety steps
    $\boldsymbol{m}_o(t) \leftarrow \textsc{GetMeasurements}(t)$
    $\boldsymbol{u}_{\text{safe}}(t+\alpha_s) \leftarrow \textsc{ShieldStep}(t, \boldsymbol{m}_o(t))$
    $\textsc{ApplyAction}(\boldsymbol{u}_{\text{safe}}(t+\alpha_s))$
    $t \leftarrow t + \alpha_s$
  end for
  success $\leftarrow \textsc{CheckSuccess}(\boldsymbol{o}(t))$
end while

motion $[t_{\text{I}}, t_{\text{F}}]$:

$$\chi_{\text{M}}(t) = \begin{cases} \chi_I(\tau;\, \boldsymbol{x}_0, \boldsymbol{u}_I(\cdot)), & \tau \in [0, t_{\text{I}}], \\ \chi_F(\tau;\, \boldsymbol{x}_I, \boldsymbol{u}_F(\cdot)), & \tau \in [t_{\text{I}}, t_{\text{F}}]. \end{cases} \tag{5}$$

At each safety timestep of duration α_{s}, the monitored trajectory (5) is verified against the task-specific safety constraints (SSM or PFL). If verification succeeds, the intended trajectory is executed $\boldsymbol{u}_{\text{safe}}(t) = \boldsymbol{u}_{\text{I}}(t)$; otherwise, the last verified failsafe is applied $\boldsymbol{u}_{\text{safe}}(t) = \boldsymbol{u}_{\text{F}}(t)$, bringing the system to an invariably safe state. We repeat this process until receiving a new action chunk from the policy after $h\Delta t/\alpha_{\text{s}}$ safety steps. Since this safeguarding mechanism is provably safe by induction [21], PACS enables the provably safe deployment of DPs and VLAs.

TABLE I

ROBOMIMIC TASK SUCCESS RATE ON 100 ROLLOUTS.

Method	Safe?	LIFT	CAN	SQUARE	Avg.
Operational space contr.	×	**1.00**	**0.99**	**0.74**	**0.91**
OFF	×	0.92	0.83	0.34	0.70
CBF [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**

VI. EVALUATION

We conduct extensive experiments in simulation and on hardware to evaluate the real-time safeguarding capabilities of our method. Our quantitative evaluation primarily aims to validate the following hypotheses:

H1 Through behavior cloning alone, DPs cannot learn a safe behavior for interaction with moving objects.

H2 With PACS, the task success rate remains close to the unsafe policy ($\pm$ 5 %) in our experiments.

H3 Computing intended trajectories from entire action chunks leads to higher task success than single-action approaches.

H4 PACS is real-time capable and does not reduce task execution speed, unless required for safety.

A. Simulation Experiments

We evaluate PACS against an unsafe operational space controller and a CBF baseline [13] on three robomimic [27] benchmark tasks (LIFT, CAN, SQUARE) integrated in human-robot-gym [28] with a dynamic spherical obstacle. We also tested a single-action version of PACS that sequentially computes new intended trajectories from each action in the chunk.

TABLE II

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**

TABLE III

BASELINE COMPARISON AND SPEED ANALYSIS FOR THE SORTING TASK.

Human	Safety Filter	Success	Duration [s]
w/ human	CBF	0.43	35.8 ± 18.8
	PACS (**ours**)	**0.80**	**32.6 ± 11.1**
w/o human	None	**0.93**	25.2 ± 3.3
	PACS (**ours**)	**0.93**	**21.7 ± 3.9**

As shown in Table I, PACS achieves significantly higher task success than the reactive CBF filter. Comparing PACS with the OFF setting confirms that path-consistent braking has no significant impact on performance (**H2**). The action chunk trajectories improve performance by 28 % over single-action approaches (**H3**).

B. Real-World Experiments

1) Tasks and Setup: We conducted hardware experiments with a Franka FR3 manipulator for three HRI tasks, visualized in Figure 4:

- SORTING: Put red blocks into a box while a person takes green blocks (SSM – no collisions allowed).
- HANDOVER: Pick a block from a human hand with bounded impact energy (PFL).
- FEEDING: Put a fork into the mouth of the user with very strict energy limits (PFL).

We deployed a temporal UNet-based DP [3] and SmolVLA [7], each trained on 50 demonstrations per task. The SmolVLA policy struggled with SORTING and FEEDING due to the high precision required; therefore, we only evaluated it on HANDOVER. We report task success rate (Success), task success while remaining safe (Safe Success), and safety violation proportion (Safety Viol.) over 30 test rollouts.

2) Results: As shown in Table II, the safeguarded policies achieve a task success rate of about 80 % on average, comparable to the nominal policies, while completely eliminating safety violations (**H1, H2**). In contrast, the nominal policies are in an unsafe state 56 % of the time, with each rollout containing violations.

For the SORTING task, PACS achieves a 37 % higher success rate than the CBF-based filter (Table III). As visualized in Figure 3, PACS keeps the policy in distribution, whereas

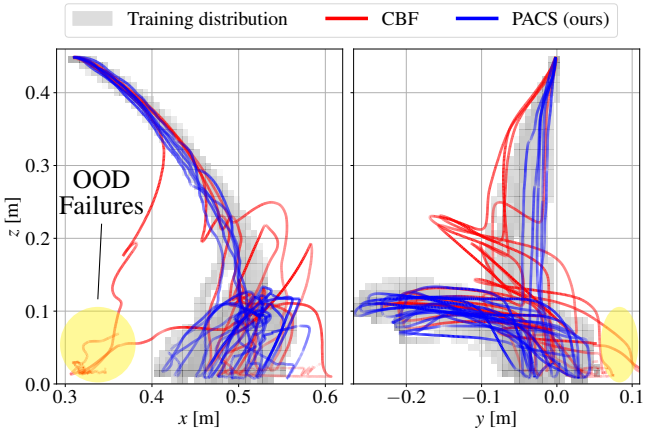
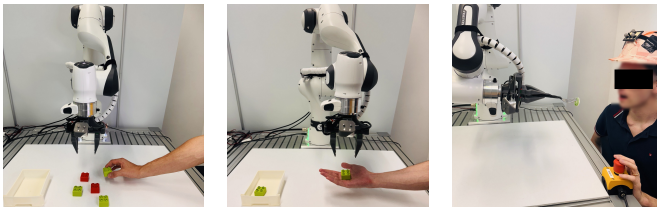


Fig. 3. End-effector paths for the SORTING task. Color intensity indicates velocity; grey areas show the training distribution. PACS reduces speed without leaving the desired path. In contrast, the CBF pushes the robot into OOD states from which the policy cannot recover.



(a) Sorting (b) Handover (c) Feeding

Fig. 4. Our three real-world tasks: SORTING, HANDOVER, and FEEDING.

the CBF frequently pushes the robot into OOD states from which recovery is not possible. Furthermore, the intermediate trajectory generation from action chunks reduces average task execution time by 14 % and increases average Cartesian end-effector speed by 13 % compared to directly sending policy actions to the low-level controller (**H4**). The average computation time of one safety step is 0.20 ms, and the intended trajectory is recomputed in about 5 ms, confirming real-time capability.

We also successfully validated PACS on the FEEDING task with a real human, highlighting 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 DPs and VLAs in dynamic environments. Our insight is that safety interventions should be path-consistent to avoid OOD 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 guarantees safety in real time on challenging HRI tasks, while reactive CBF-based filtering frequently leads policies into unrecoverable states. We consider this 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, "Denoising 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. 10 954–10 961, 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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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. (L4DC)*, vol. 283, 2025, pp. 791–803.

[25] 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.

[26] 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.

[27] 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.

[28] 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.