Efficient and Reliable Teleoperation through Real-to-Sim-to-Real Shared Autonomy

Shuo Sha[1], Yixuan Wang[1], Binghao Huang[1], Antonio Loquercio[2], Yunzhu Li[1]
[1]Columbia University [2]University of Pennsylvania

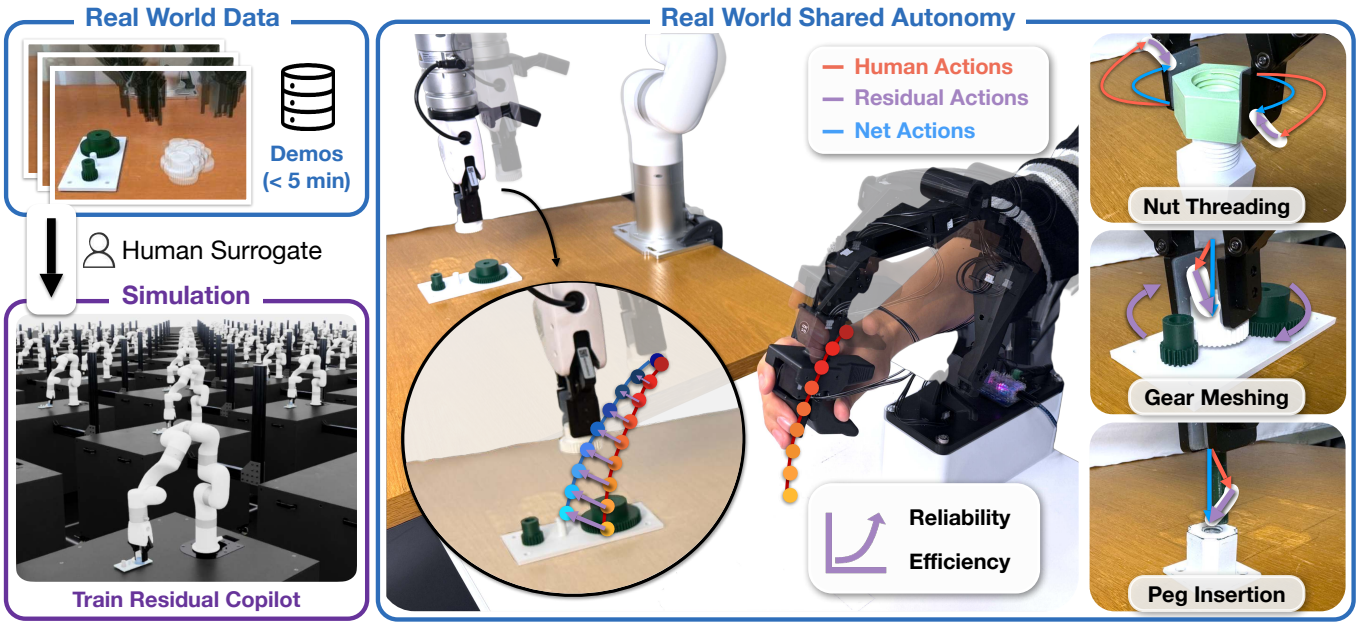

Fig. 1: **Overview of our real-to-sim-to-real shared autonomy framework.** A small amount of real teleoperation data (< 5 minutes) is used to construct a lightweight human surrogate, which drives simulation-based training of a residual copilot policy. At deployment, the copilot provides low-level corrective actions that combine with human commands to produce reliable and efficient shared autonomy for fine-grained, contact-rich manipulation tasks, where direct teleoperation struggles with precise alignment, axis-constrained rotation, and contact regulation, including nut threading, gear meshing, and peg insertion.

***Abstract*—Fine-grained, contact-rich teleoperation remains slow, error-prone, and unreliable in real-world manipulation tasks, even for experienced operators. Shared autonomy offers a promising way to improve performance by combining human intent with automated assistance, but learning effective assistance in simulation requires a faithful model of human behavior, which is difficult to obtain in practice. We propose a real-to-sim-to-real shared autonomy framework that augments human teleoperation with learned corrective behaviors, using a simple yet effective k-nearest-neighbor (kNN) human surrogate to model operator actions in simulation. The surrogate is fit from less than five minutes of real-world teleoperation data and enables stable training of a residual copilot policy with model-free reinforcement learning. The resulting copilot is deployed to assist human operators in real-world fine-grained manipulation tasks. Through simulation experiments and a user study with sixteen participants on industry-relevant tasks, including nut threading, gear meshing, and peg insertion, we show that our system improves task success for novice operators and execution efficiency for experienced operators compared to direct teleoperation and shared-autonomy baselines that rely on expert priors or behavioral-cloning pilots. In addition, copilot-assisted teleoperation produces higher-quality demonstrations for downstream imitation learning. Website: https://residual-copilot.github.io/**

I. Introduction

Teleoperation is essential for executing complex robotic manipulation tasks and collecting demonstrations to train visuomotor policies. However, for fine-grained, contact-rich manipulation, teleoperation is often slow and error-prone, making it a practical bottleneck for both deployment and scaling high-quality data.

A key reason lies in a mismatch between human capabilities and the role assigned to them in conventional teleoperation. Humans excel at high-level intent and task reasoning but struggle with continuous millimeter-scale alignment and contact regulation through a teleoperation interface, due to limited viewpoints, latency, and the embodiment gap. These low-level behaviors are better handled by an automated copilot.

Shared autonomy addresses this by assisting operators during execution. Prior methods either map human commands toward a stronger action prior such as an expert policy [3, 5, 9, 11, 18, 19], or learn a copilot conditioned on pilot commands via reinforcement learning or residual formulations [2, 12, 13, 15]. However, methods relying on expert priors shift the

challenge to obtaining the prior itself, while existing copilot-learning approaches either require extensive human-in-the-loop training, large-scale surrogate data, or remain difficult to deploy in complex real-world settings.

We propose a real-to-sim-to-real shared autonomy pipeline that sidesteps these limitations. Rather than first solving full autonomy, we learn low-level corrective behaviors that preserve human intent. Our approach trains a residual copilot using model-free reinforcement learning in simulation, driven by a lightweight k-nearest-neighbor (kNN) human surrogate fit from fewer than five minutes of real teleoperation data. By remaining within empirical data support, the kNN surrogate enables stable copilot training and effective sim-to-real transfer.

We evaluate on fine-grained, contact-rich assembly tasks, including nut threading, gear meshing, and peg insertion, through simulation experiments and a user study with novice and experienced operators. Our system improves task success for novices and execution efficiency for experienced operators compared to direct teleoperation and shared-autonomy baselines, while also producing higher-quality demonstrations for downstream imitation learning.

In summary, our contributions are: (1) a real-to-sim-to-real shared autonomy pipeline that assist teleoperation for high-precision, contact-rich manipulation; (2) a lightweight kNN surrogate, built from fewer than five minutes of data, that serves as an effective pilot model for copilot training; and (3) demonstrated improvements in both *effectiveness* and *efficiency* for novice and experienced operators, enabling higher-quality demonstration collection and practical assistive and remote teleoperation.

II. Method

A. Problem Definition

We model shared autonomy as a POMDP $\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, O, \gamma \rangle$. The full state $x_t = (s_t, g)$ comprises the environment state s_t (end-effector pose/velocity, gripper state, object poses in end-effector frame) and the pilot's latent goal g, which is unobservable to the copilot. The pilot issues a goal-implicit command $a_t^h = [p, q, u] \in \mathcal{A}$ (target end-effector pose and gripper command) via teleoperation. Rather than explicitly inferring g, we treat intent as expressed implicitly through these commands.

The copilot observes $o_t = (s_t, a_t^h)$ and learns a residual correction $\pi_r(a_t^{\text{res}} \mid o_t)$ on top of the pilot command, trained with PPO [14] without access to dynamics, goals, or the pilot policy. The reward decomposes as $\mathcal{R} = \mathcal{R}_{\text{general}} + \mathcal{R}_{\text{success}}$, where $\mathcal{R}_{\text{general}}$ captures goal-agnostic objectives (termination/contact penalties, alignment shaping, action regularization) and $\mathcal{R}_{\text{success}}$ provides a sparse task-specific success signal.

B. Human Surrogate Model

Training with human pilots in the loop is costly. Following Reddy et al. [12], Schaff and Walter [13], we train in simulation with a surrogate that must (1) remain coherent under copilot-induced distribution shift, and (2) be constructible from limited data.

We adopt a non-parametric k-nearest-neighbor surrogate π_h^{kNN} (`kNN Pilot`) built from a small demonstration set that may include unsuccessful trials. By retrieving actions directly from the empirical manifold, the `kNN Pilot` avoids extrapolation beyond distributional support, promoting stable behavior under moderate distribution shift. Neighbors are retrieved using a weighted distance over end-effector commands combining L_2 position distance, geodesic distance in $\mathrm{SO}(3)$ for orientation, and L_1 norm for the gripper command, with weights tuned on the demonstration set. We augment the surrogate with two mechanisms:

Action chunking. The surrogate retrieves k nearest neighbors, samples one via temperature-scaled softmax over negative distances, and outputs a short chunk of consecutive commands to preserve temporal consistency.

Smooth local perturbations. To expand coverage beyond empirical support, retrieved commands are composed with perturbations $\delta a_t = m_t \varepsilon_t$, where ε_t is i.i.d. uniform noise and m_t is a smooth gate (exponentially smoothed Bernoulli process) that controls gradual onset and decay of noisy phases.

C. Real-to-Sim-to-Real

A small set of real teleoperation demonstrations is used to fit the `kNN Pilot` and calibrate the simulator. We tune joint-space PID gains, task-space admittance parameters [6, 7], and physical parameters (friction, center of mass) to match real trajectories. For sim-to-real transfer, we apply domain randomization over calibrated controller parameters and object pose observations to bridge remaining gaps from controller discrepancies and pose estimation error.

III. Experiments

We evaluate our `Residual Copilot` on fine-grained, contact-rich manipulation tasks, aiming to address: (**Q1**) Can our copilot improve human operator performance in contact-rich assembly? (**Q2**) Does the `kNN Pilot` outperform prior human surrogates and guided diffusion baselines? (**Q3**) Does copilot-assisted data improve downstream imitation learning?

Experiment Setup. We evaluate through a real-world user study (Sec. III-A), a pilot model robustness analysis (Sec. III-B), and a downstream imitation learning comparison (Sec. III-C). We consider three contact-rich assembly tasks from NIST board #1 [1]: **Gear Meshing** (pick, insert, and mesh the medium-sized gear); **Nut Threading** (pick an M32 nut and tighten $\geq 180^\circ$ via three successive 60° turns); and **Peg Insertion** (pick and fully insert an $8\,\mathrm{mm}$ peg).

Participants teleoperate a UFactory xArm7 via GELLO [17] in 7-DoF task space. A wrist-mounted force–torque sensor provides measurements for admittance control, and an Intel RealSense D455 supplies RGB-D input for object pose estimation with FoundationPose [16]. Simulation experiments use Isaac Lab [8] with customized Factory [10] environments.

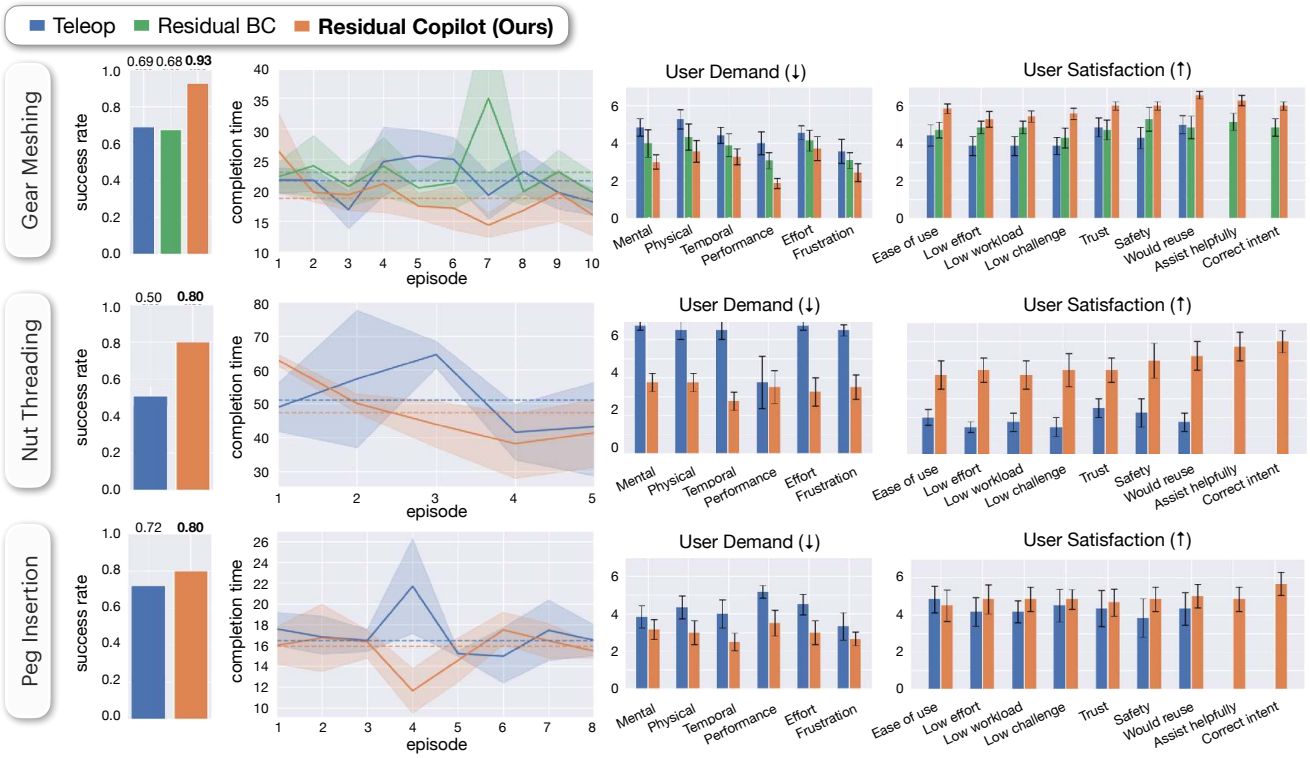

Fig. 2: **Human experiment results across three contact-rich assembly tasks.** Rows: Gear Meshing, Nut Threading, Peg Insertion. **Left:** success rate and completion time (solid line: mean; shaded: std; dashed: per-method mean). **Right:** NASA-TLX workload (lower is better) and user satisfaction (higher is better). The `Residual Copilot` consistently improves success rate and completion time while reducing workload and increasing satisfaction.

Subject Allocation. We recruited 12 novice teleoperators for gear meshing and peg insertion, and 4 experienced operators for nut threading, totaling over 20 hours of teleoperation. We interleave direct and copilot-assisted trials to control for learning effects.

Baselines. We implement the following pilot models: (i) `BC Pilot`, a Diffusion Policy (DP) [4] trained on < 5 minutes of augmented real teleoperation data per task, following Schaff and Walter [13]; (ii) `Expert-based Pilots`, comprising a near-optimal `Expert Pilot` (DP trained on 2,000 successful rollouts from our copilot), plus `Laggy` (p=0.8 action repeat) and `Noisy` (p=0.5 smooth gated noise; Sec. II-B) variants following Reddy et al. [12]. Using these pilots, we train copilot baselines: (i) `GD Copilots`, guided-diffusion copilots [18] that denoise actions conditioned on pilot commands, with `GD BC` trained on augmented teleoperation data, `GD Expert` on expert rollouts (an impractical upper bound); (ii) `Residual BC`, a residual RL copilot assisting the `BC Pilot`, differing from our method only in pilot model choice.

A. Copilot Performance

To address **Q1**, we conduct a user study comparing direct and copilot-assisted teleoperation on three contact-rich assembly tasks, evaluating objective performance and subjective experience (Fig. 2).

For **Nut Threading**, operators frequently lose alignment or apply insufficient torque, causing premature disengagement (Fig. 3). The copilot stabilizes orientation and amplifies rotational commands, improving success by up to 30%. For **Gear Meshing**, direct teleoperation fails due to imperceptible misalignments. The copilot applies subtle rotational corrections at contact, reducing the low-level alignment burden. For **Peg Insertion**, the copilot stabilizes the end-effector at an effective pre-grasp pose while leaving grasp timing to the operator, improving reliability while preserving task-level control.

Across tasks, the `Residual Copilot` consistently improves success rates and reduces completion time by decreasing failed attempts and corrective motions. These gains align with subjective reports: participants report lower NASA-TLX workload and higher satisfaction, indicating that residual assistance improves task outcomes while reducing cognitive and physical burden.

B. Human Surrogate Analysis

To address **Q2**, we compare our `Residual Copilot` against `Residual BC` with novice teleoperators on gear meshing (Fig. 2, top row). Our copilot improves both objective performance and subjective ratings, with participants reporting higher *correct intent*, suggesting the `kNN Pilot` yields more reliable corrections that better preserve human intent.

In simulation (Table I), our copilot achieves the highest in-distribution performance and smallest degradation under pilot mismatch. `Residual BC` achieves consistently lower task progression under suboptimal pilots, suggesting it fails to explore success-relevant regions under copilot-induced distribution shift with sparse rewards. `GD Copilots` are highly sensitive to pilot choice: guiding `GD Expert` with the `Expert`

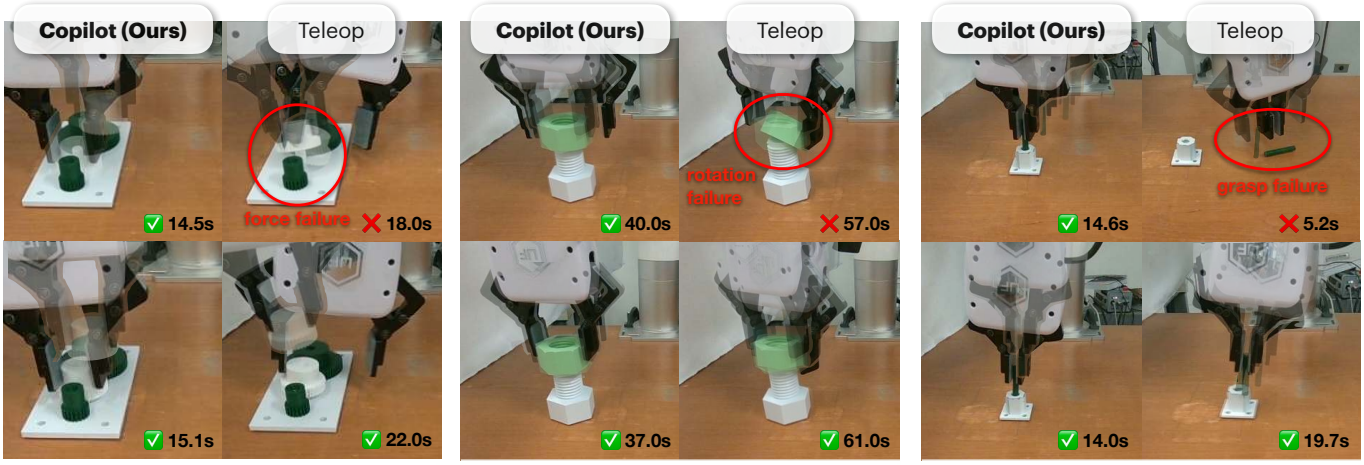

Fig. 3: **Qualitative results.** Trajectory overlays comparing unassisted teleoperation and our `Residual Copilot`. **Top (effectiveness):** the copilot enables reliable completion where teleoperation fails due to force, rotation, or grasp errors. **Bottom (efficiency):** when both succeed, the copilot reduces completion time with smoother motions.

Copilot	Eval. Pilot														
	Laggy Pilot			Noisy Pilot			Expert Pilot			BC Pilot			kNN Pilot		
	Gear	Peg	Nut	Gear	Peg	Nut	Gear	Peg	Nut	Gear	Peg	Nut	Gear	Peg	Nut
`No Copilot`	0.97	**0.94**	0.49	0.92	0.91	0.24	**0.99**	**0.99**	0.61	0.84	0.71	0.03	0.87	0.85	0.16
`GD Expert`	0.96	0.89	0.24	0.95	0.91	0.25	0.94	0.90	0.28	0.94	**0.89**	**0.34**	0.95	0.90	0.27
`GD BC`	0.63	0.56	0.00	0.65	0.56	0.00	0.62	0.53	0.01	0.63	0.52	0.01	0.66	0.55	0.00
`Residual BC`	0.96	0.44	0.00	0.91	0.44	0.00	0.98	0.44	0.00	0.70	0.40	0.00	0.86	0.46	0.02
`Residual Copilot (Ours)`	**0.99**	0.92	**0.65**	**0.96**	**0.92**	**0.69**	0.98	0.93	**0.74**	**0.97**	0.83	0.21	**1.00**	**0.99**	**0.81**

TABLE I: **Cross-pilot generalization in simulation.** Rows: training pilot; columns: evaluation pilot across three tasks. Each cell reports task progression over 1,000 trials. Diagonal entries (gray/blue) denote in-distribution evaluation; off-diagonal entries measure generalization to unseen pilots. The `Residual Copilot` maintains consistently high performance across evaluation pilots, demonstrating robustness to pilot mismatch and perturbations.

`Pilot` *reduces* performance relative to `No Copilot`, while steering `GD BC` with stronger priors yields no improvement, indicating that test-time guidance alone cannot reconcile action optimality with preservation of human intent.

C. Data Quality Comparison

To address **Q3**, we train a vision-based Diffusion Policy on gear-meshing data from five participants under two conditions: (i) **matched attempts**, using all successes from equal attempts per method, and (ii) **matched successes**, using the same number of successful demonstrations.

Copilot-assisted data yields substantially higher downstream performance under both conditions (Table II). Matched successes isolates demonstration *quality*: copilot assistance produces more consistent trajectories in alignment, contact timing, and approach strategy (Fig. 3), whereas teleoperation demonstrations contain larger corrective motions that increase variability. These results highlight shared autonomy as a tool for scalable, high-quality demonstration collection.

Method	Matched attempts		Matched successes	
	Grasp	Insert	Grasp	Insert
Teleop	7/20	1/20	6/20	0/20
Residual (Ours)	**18/20**	**11/20**	**19/20**	**9/20**

TABLE II: **Downstream policy progression from copilot vs. teleoperation data (Gear Meshing).** Diffusion policies trained on copilot-assisted or direct teleoperation demonstrations, controlling for data budget (matched attempts) and success count (matched successes). Copilot assistance improves learnability and consistency of collected demonstrations.

IV. Conclusion

We presented a real-to-sim-to-real shared autonomy framework for fine-grained, contact-rich teleoperation. Our approach trains a residual copilot using reinforcement learning in simulation, driven by a lightweight kNN human surrogate fit from fewer than five minutes of real teleoperation data. By learning only low-level corrective behaviors on top of human commands, the copilot preserves operator intent while avoiding reliance on brittle parametric behavioral-cloning pilots or expert autonomous priors.

Across both simulation and a real-world user study on nut threading, gear meshing, and peg insertion, our system improves task success for novices and execution efficiency for experienced operators relative to direct teleoperation and prior shared-autonomy baselines. Beyond execution performance, copilot-assisted teleoperation produces demonstrations with greater consistency and structure, leading to improved downstream imitation learning policy performance.

References

[1] Assembly performance metrics and test methods. https://www.nist.gov/el/intelligent-systems-division-73500/robotic-grasping-and-manipulation-assembly/assembly, April 2022.

[2] Kal Backman, Dana Kulić, and Hoam Chung. Reinforcement learning for shared autonomy drone landings. *Auton. Robots*, 47(8):1419–1438, October 2023. ISSN 0929-5593. doi: 10.1007/s10514-023-10143-3. URL https://doi.org/10.1007/s10514-023-10143-3.

[3] Alexander Broad, Todd Murphey, and Brenna Argall. Learning models for shared control of human-machine systems with unknown dynamics, 2018. URL https://arxiv.org/abs/1808.08268.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

[5] Yunxin Fan and Monroe Kennedy III. Diffusion-safe: Shared autonomy framework with diffusion for safe human-to-robot driving handover, 2025. URL https://arxiv.org/abs/2505.09889.

[6] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control, 2025. URL https://arxiv.org/abs/2410.09309.

[7] Cunqiu Liu, Junyao Gao, Yi Liu, Xuanyang Shi, Fangzhou Zhao, Jingchao Zhao, and Chuzhao Liu. Admittance control of manipulators in unknown environment. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, page 2157–2162. IEEE Press, 2018. ISBN 978-1-5386-6074-4. doi: 10.1109/ICMA.2018.8484383. URL https://doi.org/10.1109/ICMA.2018.8484383.

[8] Mayank Mittal, Pascal Roth, James Tigue, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

[9] Rocco Moccia, Cristina Iacono, Bruno Siciliano, and Fanny Ficuciello. Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery. *IEEE Robotics and Automation Letters*, 5(2):1650–1655, 2020. doi: 10.1109/LRA.2020.2969941.

[10] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, Ankur Handa, and Dieter Fox. Factory: Fast contact for robotic assembly, 2022. URL https://arxiv.org/abs/2205.03532.

[11] Eley Ng, Ziang Liu, and Monroe Kennedy III. Diffusion co-policy for synergistic human-robot collaborative tasks, 2023. URL https://arxiv.org/abs/2305.12171.

[12] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning, 2018. URL https://arxiv.org/abs/1802.01744.

[13] Charles Schaff and Matthew R. Walter. Residual policy learning for shared autonomy. In *Robotics: Science and Systems (RSS)*, 2020. URL https://arxiv.org/abs/2004.05097.

[14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[15] Weihao Tan, David Koleczek, Siddhant Pradhan, Nicholas Perello, Vivek Chettiar, Vishal Rohra, Aaslesha Rajaram, Soundararajan Srinivasan, H M Sajjad Hossain, and Yash Chandak. On optimizing interventions in shared autonomy, 2022. URL https://arxiv.org/abs/2112.09169.

[16] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

[17] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2024. URL https://arxiv.org/abs/2309.13037.

[18] Zhao-Heng Yin, Changhao Wang, Luis Pineda, Francois Hogan, Krishna Bodduluri, Akash Sharma, Patrick Lancaster, Ishita Prasad, Mrinal Kalakrishnan, Jitendra Malik, Mike Lambeta, Tingfan Wu, Pieter Abbeel, and Mustafa Mukadam. Dexteritygen: Foundation controller for unprecedented dexterity, 2025. URL https://arxiv.org/abs/2502.04307.

[19] Takuma Yoneda, Luzhe Sun, , Ge Yang, Bradly Stadie, and Matthew Walter. To the noise and back: Diffusion for shared autonomy, 2023. URL https://arxiv.org/abs/2302.12244.