Uncertainty Comes for Free: Human-in-the-Loop Policies with Diffusion Models

Zhanpeng He*
Department of Computer Science
Columbia University
New York, New York 10027
Email: zhanpeng@cs.columbia.edu

Yifeng Cao*
Department of Electrical Engineering
Columbia University
New York, New York 10027
Email: yc4317@columbia.edu

Matei Ciocarlie
Department of Mechanical Engineering
Columbia University
New York, New York 10027
Email: matei.ciocarlie@columbia.edu

***Abstract*—Human-in-the-loop (HitL) robot deployment has gained significant attention in both academia and industry as a semi-autonomous paradigm that enables human operators to intervene and adjust robot behaviors at deployment time, improving success rates. However, continuous human monitoring and intervention can be highly labor-intensive and impractical when deploying a large number of robots. To address this limitation, we propose a method that allows diffusion policies to actively seek human assistance only when necessary, reducing reliance on constant human oversight. To achieve this, we leverage the generative process of diffusion policies to compute an uncertainty-based metric based on which the autonomous agent can decide to request operator assistance at deployment time, without requiring any operator interaction during training. Additionally, we show that the same method can be used for efficient data collection for fine-tuning diffusion policies in order to improve their autonomous performance. Experimental results from simulated and real-world environments demonstrate that our approach enhances policy performance during deployment for a variety of scenarios.**

I. Introduction

Human-in-the-Loop (HitL) operation is a paradigm where a human operator can intervene and assist a robot during deployment. This paradigm is seeing increasing adoption in cases where robots must continue to operate adequately even in corner cases not foreseen before deployment.

In parallel, even as recent advances in policy learning have shown significant improvements in robustness at deployment time [6, 8, 16, 19], current methods can still fail due to problems such as data distribution shift [14] or incomplete state observability [7]. To address this issue, HitL methods can be a natural fit for learning agents: the robot operates autonomously when possible, leveraging the ability of policy learning to execute complex motor control tasks. An expert operator can take over for corner cases, ensuring task success. However, deploying HitL can be labor-intensive and impractical if it implies constant monitoring of the robot's behavior by the human operator, or frequent interventions.

In this work, we propose a data-driven approach for generating HitL policies. We start from the basic HitL premise: the robot generally acts autonomously, but a human operator is available to provide teleoperation commands should the robot require them. Our method is designed to determine when the agent should request such expert assistance, making effective use of a limited number of such calls during deployment. We also remove the need for expert intervention during the training phase, as that would place a large burden on the operator. This means that the agent has no knowledge about the effect of assistance, except for the assumption that is effective for task completion.

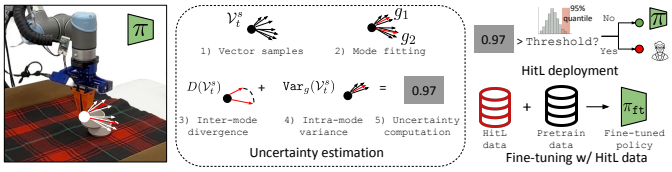

Fig. 1: **HitL policies with denoising uncertainty:** We propose using denoising uncertainty as a metric for deciding when to request (human) expert assistance. Predicted de-noising vectors in end-effector position space (illustrated here via arrows on end-effector position) are collected in a vector field, whose inter-mode divergence and intra-mode variance are used to compute policy uncertainty; when this measure exceeds a threshold, operator assistance is requested. We also show that ensuing teleoperation data can be used to fine-tune policies, achieving notable performance improvements with minimal additional data.

To achieve this, we utilize diffusion models [6] as our policy class. Diffusion policies offer two key advantages: (1) they have demonstrated robust performance in imitation learning tasks, and (2) their generative process involves an iterative denoising mechanism which we can leverage for insight into the agent's decision-making process. Specifically, we use the denoising information to compute an uncertainty metric for the policy, which is then used during deployment to determine when human intervention is most beneficial (Fig. 1). To achieve this, we directly leverage the noise prediction model learned during the policy training process. Therefore, uncertainty estimation does not require training any additional models, imposes a minimal cost at run time, and can thus be considered a "free" byproduct of diffusion policy training. Finally, we show that data collected during the operator interventions can be incorporated back into training through a fine-tuning process that further improves policy performance. Our main contributions are as follows:

- We propose a novel method for HitL policy execution using uncertainty estimation with diffusion policies. Our method does not require human-robot interaction during

*Equal contributions.

training, and incurs minimal computational overhead during deployment.
- We validate our method across multiple types of deployment challenges, in both simulated and real environments. Experimental results show that our approach requires fewer human interventions and achieves higher task performance compared to alternative learning-based HitL agents.
- We also show that our uncertainty-based state identification method can be utilized to collect targeted fine-tuning data, yielding performance improvements with smaller datasets compared to collecting additional full-trajectory demonstrations.

II. Related Work

Our work is closely related to interactive imitation learning (IIL), where a learning agent queries an expert for additional labels during policy execution and augments the training dataset with expert demonstrations [21, 15, 18, 24, 5, 17]. An effective data collection strategy for IIL is human-gated DAgger [12], which relies on a human to continuously monitor and intervene during robot execution. However, such continuous supervision is inefficient and undermines the goal of robot learning. To make HitL systems practical, the robot must be strategic in when and how it requests human assistance. Prior work has explored budgeting expert queries in IIL. For example, Hoque et al. reduce human effort by constraining robot queries using models of limited human attention [10] or fixed intervention budgets [9]. However, these methods require extra training procedures during training or deployment, increasing computational demands. This overhead is especially problematic during policy execution, where it may introduce additional latency. In contrast, our approach avoids additional training and supports efficient parallelization during deployment, resulting in minimal runtime overhead. Other approaches leverage action consistency [1], diffusion loss [25], or online conformal prediction [27] to decide when to query the human. One crucial aspect of this line of work is the selection or online-tuning of their thresholds. To minimize human efforts in a HitL pipeline, our method leverages the multi-modal nature of human demonstrations to detect critical states where the robot is uncertain. We demonstrate that our method is robust to threshold selection in different scenarios.

III. Method

We begin the description of our method with a short recap of diffusion policies, specifically the action denoising process which we will then leverage to introduce our metric for quantifying uncertainty. After that, we describe how this metric can be used by a HitL diffusion policy agent to determine when to request operator assistance, and finally how teleoperation data obtained through this method can in turn be used to fine-tune the original policy.

Diffusion policies generate actions through an action-denoising process, leveraging denoising diffusion probabilistic models (DDPM). A DDPM models a continuous data distribution $p(a^0)$ as reversing a forward noising process from a^0 to a^K, where a^K is Gaussian noise sampled from $\mathcal{N}(0, \sigma^2\mathcal{I})$. The generative process $\pi(\mathrm{a_t}|\mathrm{o_t})$, where $\mathrm{o_t}$ and $\mathrm{a_t}$ are robot observations and actions at time step t, starts by sampling an action a_t^K as random noise, and then iteratively denoises it using:

$$a_t^{k-1} = \beta(a_t^k - \gamma\epsilon_\theta(o_t, a_t^k, k) + \mathcal{N}(0, \sigma^2\mathcal{I})) \quad (1)$$

where β, σ and γ are functions of iteration step k. $\epsilon_\theta(o, a^k, k)$ is a learned model that predicts the noise to be removed at each denoising step.

A. Denoising-based uncertainty metric

To estimate the uncertainty of a diffusion-based agent, our method leverages the generative process described above. In particular, we assume that our diffusion policy is operating on task space control, which is a very common case in recent diffusion-based robot policy learning methods [20, 26, 11], and outputs absolute end-effector poses as part of its action vector. In this case, the noise predicted (and removed) during the generative process can be interpreted as a vector field pointing toward the distribution for intended end-effector pose the at the next step. We can thus leverage this vector field to analyze whether the diffusion-based agent is confident about its generative target.

Our goal is to estimate an uncertainty metric $\texttt{Uncertainty}(o_t)$ where o_t is the observation at time step t. We begin by sampling a set of end-effector poses $\mathcal{A}_t^s$, where each entry $a_t^s \in \mathcal{A}_t^s$ is within a distance r from the current pose. When operating in task-space position control, each of these samples can be interpreted as an action vector. We can thus feed these samples through the diffusion policy noise prediction model, and collect the predicted noise vectors: let the set $\mathcal{V}_t^s$ contain all vectors $v_t^s = \epsilon_\theta(o_t, a_t^s, 0)$ computed for each $a_t^s \in \mathcal{A}_t^s$. This vector field encodes directions toward the action distribution that the policy aims to recover. We will use these denoising vectors to estimate uncertainty, defined as $\texttt{Uncertainty}(o_t) = f(\mathcal{V}_t^s)$.

The simplest method to assess uncertainty is to consider the variance of the vector field $\mathcal{V}_t^s$. However, diffusion policies are often used for their ability to capture multi-modality in the underlying demonstrations: from any given state, there might be multiple distinct action trajectories that accomplish the desired task. Thus, the denoising vector field could reflect the multi-modal nature of the demonstration data, and naive variance estimation of the vector field may fail to capture this effect.

To address this, we use Gaussian Mixture Models (GMMs) to capture the potentially multi-modal nature of action generation. Our method starts by fitting the collected denoising vectors with N GMMs, each using a different number of modes. We then select the best-fit GMM for uncertainty estimation via maximum likelihood estimation:

$$\max_{n,\theta_\mathrm{g}} P(\mathcal{V}_t^s; n, \theta_\mathrm{g}),$$

where n is the number of modes and θ_g contains the parameters of the GMM. With the best-fit GMM, we then estimate the agent's uncertainty. We first evaluate the divergence between each mode:

$$D(\mathcal{V}_t^s) = \frac{1}{n(n-1)} \sum_{i,j} 1 - S_c(g_i, g_j)$$

where,

$$S_c(g_i, g_j) = \frac{g_i \cdot g_j}{||g_i|| \cdot ||g_j||}$$

Here, g_i represents the mean of the i^{th} mode and S_c represents cosine similarity between two vectors. We also evaluate the GMM variance as part of the uncertainty estimation:

$$\texttt{Var}_g(\mathcal{V}_t^s) = \sum_i p(v_i)\texttt{Var}(v_i) \tag{2}$$

where $\texttt{Var}$ represents the variance of vector data and v_i represents vector samples belongs to the i^{th} mode of the GMM. Putting them together, we can estimate the overall uncertainty as:

$$\texttt{Uncertainty}(o_t) = D(\mathcal{V}_t^s) + \alpha \texttt{Var}_g(\mathcal{V}_t^s), \tag{3}$$

where α is a constant. This uncertainty estimation considers two aspects during denoising: how diverged the target distributions are, and how much entropy there is in each of the modes.

B. Uncertainty-based intervention and policy fine-tuning

Having defined our uncertainty metric, we can use it during deployment by setting a threshold to determine whether we to request human assistance. At every state, the agent computes its own uncertainty and, if the level of uncertainty exceeds the threshold, the agent requests that the operator take control and teleoperate the system for several steps, until uncertainty returns below the threshold.

In addition, our method can also be used to collect data to further fine-tune the policy. This allows for better performance in the next policy execution. To fine-tune a policy, we save the observation and action pairs $\{\mathrm{O}, \mathrm{A}\}$ when a human operator is intervening with the robot and use this data set to fine-tune the underlying diffusion policy. To avoid catastrophic forgetting [2], we sample from both the fine-tuning dataset $\mathcal{D}_{ft}$ and pretraining dataset $\mathcal{D}_{train}$. For each mini-batch, we ensure 50% are from $\mathcal{D}_{ft}$. Our approach implicitly means that this fine-tuning data specifically addresses the areas of state space where the agent's uncertainty is high, since that is where operator assistance is requested.

Putting all components together, the method contains three main steps: 1. train a diffusion policy; 2. deploy the policy, and request operator control if policy uncertainty estimated by our metric exceeds a preset threshold; 3. (optional) use human intervention data to fine-tune the diffusion policy.

IV. Experiments

To test applicability of this framework, we consider three types of deployment issues that typically cause uncertainty for learning-based agents. **Case 1: Data distribution shift**, such as visual observation distribution shift caused by change of lighting conditions, or a change in environment dynamics due to interaction with novel objects. **Case 2: Incomplete state observability**, commonly approached by redesigning, adding or moving sensors, but difficult to tackle in the general case. **Case 3: Incorrect choice between different action modes**, where the agent is presented with a discrete choice between two or more action trajectory modes equally well represented during training. While diffusion policies are naturally well-equipped to make such choices, task under-specification can lead to the selection of the incorrect action mode for the given goal.

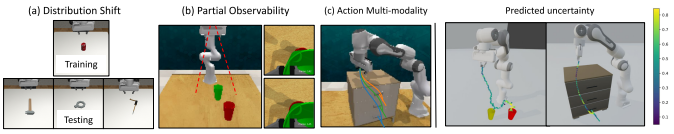

Fig. 2: **Experiments in simulated environments.** Left: we consider three scenarios during policy deployment. (a) Distribution shift; (b) Partial observability (c) Action multi-modality. Right: qualitative visualization of predicted uncertainty, with lighter colors indicating higher uncertainty.

During policy execution, these problems may not be present in all states – many states are easy to make decisions for, and require no human intervention (e.g. moving the arm in free space). The goal of our metric is to identify when the issues described above arise, and selectively request help. For Case 3 above, we posit that a few steps under teleoperator control can "steer" the policy towards the desired mode, after which autonomous operation can resume. Case 1 lends itself well to fine-tuning based on the novel data collected during teleoperation. Finally, we expect Case 2 to be the most difficult, since correct decision making is impossible without changing the available observation. We design our experiment set to test a range of scenarios covering these situations.

A. Evaluation and Baselines

We validate our method across the three types of deployment challenges in both simulated and real environments. In our test scenarios, full teleoperation generally succeeds, with sufficient human intervention achieving near 100% task success. However, a key goal of HitL deployment is efficiency: assistance should be requested conservatively to minimize unnecessary interruptions.

Our evaluation thus focuses on two core aspects. First, we measure the efficiency of human-robot interaction by tracking the required frequency of human interventions to achieve 100% task success with K rollouts (K depends on tasks and $K \geq 20$). Second, we assess the improvement in task performance enabled by human assistance and policy fine-tuning, quantifying the impact of integrating human feedback. We compare our approach against three state-of-the-art baselines that incorporate uncertainty estimation into HitL frameworks:

- **`ThriftyDAgger`** [9], which uses a model-ensemble-based "novelty" (i.e., out-of-distribution) detection and a risk metric learned via Bellman updates on test-time data.
- **`Diff-DAgger`** [25], which uses predicted actions from a diffusion policy to compute the diffusion loss, and, based on it, a metric to decide requesting human assistance.

	Lift-sim	Cup-stacking	Open-drawer
HULA-offline [22]	55.7(±6.1)	54.0(±16.3)	21.7(±11.7)
ThriftyDAgger [9]	33.5(±7.4)	21.2(±15.6)	17.2(±8.9)
Diff-DAgger [25]	30.2(±1.3)	32.0(±4.0)	16.0(±4.4)
Our method	**16.9(±4.5)**	**5.4(±1.0)**	**8.0(±1.9)**
Avg. Full-traj. Length	76.6(±5.9)	147.8(±12.9)	114.8(±5.7)

TABLE I: Average # of human assistance steps needed to achieve 100% success rates for simulated tasks.

- **HULA-offline** [22], which produces an RL-based HitL policy by explicitly estimating the variance of state values. We adapt it to offline RL by implementing an offline variant using Conservative Q-Learning (CQL) [13].

B. Simulated Environments

We now summarize the simulated environments used to test our method. **Distribution shift: Lift-sim.** In this task, we ask the robot to grasp and lift objects in a table-top setting. To emulate distribution shift, demonstration data is collected using only a single object (red cube - see Fig.2), while for testing we roll out the pretrained policy to a set of unseen objects (round nuts, hammers, and hooks). **Partial observability: Cup Stacking.** Here, we ask the robot to grasp a green cup and place it inside a red cup. We use three views as our observation: front, side, and wrist. Successful execution requires the robot to infer object alignment based on its observations. Misalignment can lead to unintended collisions, leading to failures. To introduce variability, cup positions are randomized during data collection. **Incorrect choice of action mode: Open drawer.** Here, the robot is tasked with opening one of three drawers in the scene. The collected dataset includes trajectories for opening each drawer, with $1/3$ of the data corresponding to each drawer. However, the dataset does not specify which drawer is to be opened in a given trajectory, introducing under-specification.

As a sanity check, we first evaluate the unassisted task performance of the diffusion policy on each task under the training data distribution. For the Lift-sim task, the fully autonomous policy achieves 100% success rate on the training object but fails completely (0% success rate) on unseen objects. For Cup Stacking, the robot consistently picks up the first cup (100% grasp rate) but fails to place it into the second cup due to alignment difficulties, resulting in a success rate of 0% without human assistance. This task is also sensitive to observation selection – training with only side and front views causes the robot to fail when grasping the green cup. For Open Drawer, the fully autonomous policy learns to open a drawer with 100% success if the task description does not specify which particular drawer should be opened. Interestingly, despite the under-specified training (i.e., no conditioning on which drawer to open), the policy captures the multi-modality of the training distribution. During 100 rollouts with random sampling, the robot opens the middle and bottom drawers in 15% and 85% of trials, respectively, but never opens the top drawer.

C. Efficiency of Human Interactions

We now evaluate HitL deployment performance of these tasks. We note again that 100% success rate is always possible with sufficient human assistance. Thus, we focus here on achieving high success rates with as few human assistance steps as possible, which is a critical aspect for real-world scalability of HitL systems.

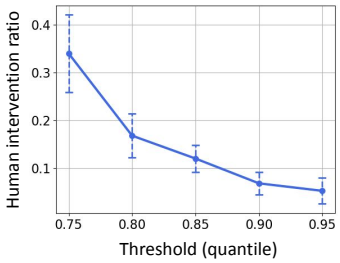

Fig. 3: Sensitivity to threshold selection.

As shown in Table I, for all simulated tasks, our method outperforms all baselines, and allows the policy to achieve perfect task success with the fewest intervention steps. Qualitatively (see accompanying video), we observe that, for Lift-sim, the robot only seeks human assistance when its gripper is close to the object, and lifting happens without intervention. For Cup Stacking, our method identifies states where the agent aligns the two cups as having high uncertainty, whereas picking up a cup (which benefits from unoccluded view) is marked as low uncertainty. Finally, for Open Drawer, the policy asks for assistance when it needs to decide which drawer to reach to, and, once the human operator steers it towards the intended target, the robot autonomously completes the rest of the task.

Looking at baselines, we find that **ThriftyDAgger** provides good uncertainty estimation for in-distribution data (e.g. high state novelty when close to the first cup and when placing on the second cup for Cup Stacking), but its autonomous behavior is less effective and thus requires a low threshold for human assistance, leading to more interventions for a 100% success rate. **Diff-DAgger**, which, like us, relies on diffusion models as a policy class, also requires more human assistance to achieve 100% success rates. Finally, **HULA-offline** performs the worst, likely due to its inability to utilize a sparse reward in an offline setting.

Several key hyperparameters influence the performance of our system. We focus here on the most critical one – the uncertainty threshold. In this work, we always use an uncertainty metric threshold set at the 95% quantile of a held-out validation set not used in training or testing. We find that this selection consistently leads to 100% success rate with low teleoperator involvement. Figure 3 shows the effect of further lowering this threshold for the cup stacking task, which, as expected, leads to more human assistance. We note that all points shown in this figure represents 100% success rate over five rollouts (cup location randomized), which highlights that our metric can consistently detects critical states that requires human assistance.

D. Fine-tuning Performance

Our method requests operator assistance in states where the policy exhibits high uncertainty. We posit that these states are particularly valuable as they highlight areas where the policy

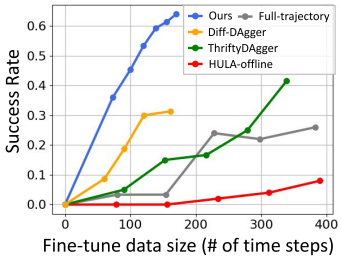

Fig. 4: Average success rate of fine-tuning the Lift-sim task with different number of human intervention steps.

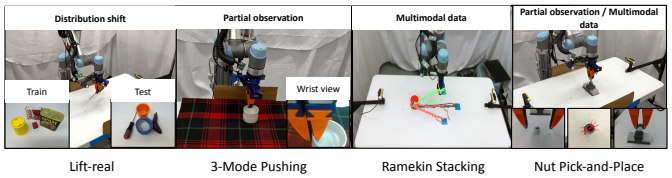

Fig. 5: **Real robot experiments:** we design our experiments to elicit the challenges described in Sec. IV on a real robot. Examples of unassisted, HitL, fine-tuned policy rollouts can be found in the supplementary video.

can benefit from additional data collection for fine-tuning. We test this hypothesis by checking if leveraging our uncertainty metrics reduces the amount of data required for fine-tuning, while still achieving significant performance improvements.

Figure 4 shows autonomous policy performance improvements as a function of the size of the fine-tuning dataset, for our method as well as the baselines. Our method also consistently achieves higher success rates with the similar amount of fine-tuning data. Among all baselines, `Diff-DAgger` shows best improvement with small amount of data. We explain this by its use of diffusion-based policies. We also note that fine-tuning on carefully curated data also outperforms the simple baseline of fine-tuning using full trajectories (i.e. complete additional demonstrations on testing scenarios). We note that the data used for each fine-tuning experiment is collected independently (i.e. the HitL fine-tuning dataset is not a part of the full-trajectory data set). For the HitL fine-tuning, the fine-tuning dataset only consists of actions when the robot is operated by human operators, instead of full trajectories.

E. Real robot experiments

Finally, we validate our method on real robot data collected via tele-operation. To support real-world deployment, we employ denoising diffusion implicit models (DDIM) [23] for high-frequency action generation. We evaluate our method on 4 real robot tasks (see Fig. 5). As in the simulated experiments, we show an example in each of the deployment problems.

	Lift-real	Stack-ing	3-Mode Pushing	Nut-PnP
`Our method`	7.2	6.8	6.5	8.4
`Avg. Full-traj. Length`	80.0	111.9	98.9	48.8

TABLE II: Average # of human assisted steps needed to achieve 100% success rate during policy deployment.

	Train	Test	$\|\|\mathcal{D}_{ft}\|\|$
Zero-shot	1	0.16	0
HitL fine-tuning (Ours)	1	0.63	80
Full-traj. fine-tuning	1	0.31	132

TABLE III: Fine-tuning performance of the Lift-real task. Results are success rates derived by 20 policy rollouts per object.

With HitL deployment, the robot can complete all four tasks. On average, our method only requests help from the human for approximately 8.3% of time steps for an 100% success rate (see Table II). We note that, since we are using action chunking during real robot deployment, one human intervention allows the human to control the robot for four steps, the same as the diffusion policy.

Qualitatively, our method identifies crucial states during policy execution. For example, in the Lift-real task, the robot asks for assistance when the gripper is close the the object. Using human-collected data with uncertainty, we can fine-tune the diffusion policy to improve 47% success rate on average (shown in Table III), outperforming fine-tuning with full-trajectories of data. In the 3-Mode Pushing task, the robot autonomously reaches to the side of the object and then transfers control to the human operator, who poses the gripper in the correct location depending on the intended target.

Once the pose of the gripper is indicative of the desired target, uncertainty drops, and the robot takes over and completes the task autonomously. In the Ramekin Stacking task, our method identifies high-uncertainty alignment states when the bottom ramekin is visually occluded (see Fig. 5). In contrast, grasping the first ramekin – where visual observations suffice – is marked as low-uncertainty and thus performed autonomously.

Finally, in the Nut Pick-and-Place task, our method assigns high uncertainty to two critical stages of execution: positioning for grasping (where the dataset contains diverse strategies for aligning the gripper with the nut edge as shown in Fig. 5) and placement (where precise positioning of the nut is required). The visual observations from the wrist camera and the two side cameras fail to reliably determine the stability of the placement, resulting in elevated uncertainty. The agent thus requests operator assistance for task completion.

V. Conclusions

In this work, we propose a novel method that enables robots to actively and efficiently request HitL assistance during deployment. Leveraging an uncertainty metric derived from the denoising process of diffusion policies, our approach identifies states where human intervention is most beneficial, thereby minimizing unnecessary monitoring and intervention. For future work, our approach can be extended to Vision-Language-Action (VLA) models that employ diffusion-based action heads [4, 3]. We also aim to investigate how to design interpretable feedback mechanisms that allow robots to express uncertainty and intent in a way that is intuitive for human operators—potentially leveraging Vision-Language Models, as demonstrated in [1]. These efforts aim to bridge the gap between fully autonomous systems and HitL deployment, fostering more efficient and scalable solutions for real-world robotic applications.

References

[1] Christopher Agia, Rohan Sinha, Jingyun Yang, Zi-ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. *arXiv preprint arXiv:2410.04640*, 2024.

[2] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[3] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 783–792. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/brown19a.html.

[6] 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. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[7] Yang Cong, Ronghan Chen, Bingtao Ma, Hongsen Liu, Dongdong Hou, and Chenguang Yang. A comprehensive study of 3-d vision-based robot manipulation. *IEEE Transactions on Cybernetics*, 53(3):1682–1698, 2021.

[8] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[9] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *5th Annual Conference on Robot Learning*, 2021.

[10] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, Karthik Dharmarajan, Brijen Thananjeyan, Pieter Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *Conference on Robot Learning*, pages 368–380. PMLR, 2023.

[11] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[12] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[13] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[14] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[15] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.

[16] Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969. IEEE, 2024.

[17] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *International conference on machine learning*, pages 2285–2294. PMLR, 2017.

[18] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

[19] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems*, 2024.

[20] Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

[21] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[22] Siddharth Singi, Zhanpeng He, Alvin Pan, Sandip Patel, Gunnar A. Sigurdsson, Robinson Piramuthu, Shuran Song, and Matei Ciocarlie. Decision making for human-in-the-loop robotic agents via uncertainty-aware reinforcement learning. In *International Conference on Robotics and Automation*, pages 7939–7945. IEEE, 2024.

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon.

Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[24] Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *16th Robotics: Science and Systems, RSS 2020*. MIT Press Journals, 2020.

[25] Yen-Ling Kuo Sung-Wook Lee, Xuhui Kang. Diff-dagger: Uncertainty estimation with diffusion policy for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[26] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[27] Michelle D Zhao, Henny Admoni, Reid Simmons, Aaditya Ramdas, and Andrea Bajcsy. Conformalized interactive imitation learning: Handling expert shift and intermittent feedback. In *The Thirteenth International Conference on Learning Representations*, 2025.