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

Author Names Omitted for Anonymous Review. Paper-ID 561

***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.**

***Index Terms*—Human-in-the-loop policies, policy fine-tuning.**

I. Introduction

Human-in-the-Loop robot deployment is a paradigm where a human operator can intervene and assist the robot during deployment. It has gained more usage in the field of robot learning due to the difficulties of deploying learned models, especially control policies, in the physical world. Although recent advances in policy learning has shown significant improvements in robustness during deploy time, current methods are still limited to constrained environments due to the lack of robotics data. Even in the fields with massive data sets (e.g. natural language processing and computer vision), large models can still make errors in simple inference tasks due to hallucinations [13]. This issue is particularly pronounced in robotics, where available datasets are far smaller compared to other domains. In robotics, hallucinations in action generations can produce catastrophic results (e.g., breaking the robots and human environment). To enable robots to operate effectively in human environments, it is crucial for them to possess the capability to collaborate with humans, especially to mitigate or recover from decision-making errors.

To address this issue, researchers have explored leveraging human input to enhance policy quality. The core idea is to provide a small amount of high-quality data to align policies with practical task requirements. Several key directions have been investigated. One prominent approach involves human-in-the-loop policies, a class of control strategies that prioritizes not only task performance but also performance improvement through human assistance.

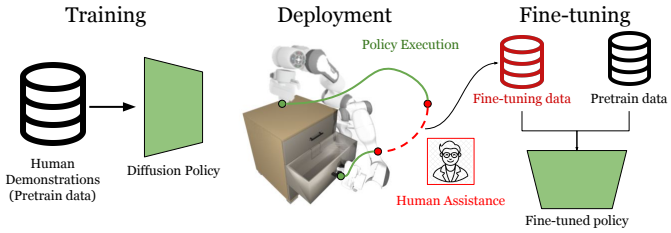


Fig. 1. **Human-in-the-loop policies:** We assume a human-in-the-loop robotic agent that actively requests human assistance when necessary. In our method, the robot uses an underlying diffusion policy for autonomous operation. During deployment, we leverage the denoising process inherent to diffusion models in order to identify states with high uncertainty (highlighted in red), where the robot seeks operator intervention. We show that this approach allows for task improvement with a small number of such operator calls. Furthermore, the human-operated data can also be used to fine-tune the policy, improving the robot's autonomous execution performance in future deployment.

However, deploying human-in-the-loop (HitL) policies can be labor-intensive due to the need for constant monitoring of robot behavior and frequent, interruptive interventions. This makes HitL deployment both impractical and costly in many applications. Previous works [31] have explored the use of uncertainty estimation within reinforcement learning (RL) frameworks as a metric to determine when human intervention is necessary. While promising, these approaches often suffer from instability and high sensitivity to the scale of rewards [10], as they depend on learning a value function from reward signals that can vary significantly across tasks and environments.

Recent works have adopted data-driven approaches for policy learning to circumvent the challenges posed by reward engineering. These approaches typically involve fitting a model to human-collected data using supervised learning. However, since human factors are often not considered during data collection, these models primarily focus on replicating or generating the training dataset, rather than accounting for human interaction.

In this work, we propose a data-driven approach for generating human-in-the-loop policies. Our method leverages diffusion models and eliminates the need for additional computation during training. Instead, we utilize the intrinsic properties of diffusion models – specifically, the denoising process. This process allows the agent to compute its internal uncertainty during deployment. When the uncertainty exceeds a certain threshold, the agent proactively seeks human assistance. The uncertainty can arise from various sources, and in this work, we specifically investigate three key causes: data distribution shifts,

partial observations, and misalignment between generated actions and user intents.

While distribution shifts and partial observability are well-known challenges in robot learning [18], generative misalignment has emerged as a new bottleneck when training control policies with offline, human-collected data. This issue arises from the inherent diversity in the data. For example, humans may use different motions to achieve the same task. However, because the model is only provided with the same task specification, capturing this diversity becomes a significant challenge for training policies. Even models that effectively capture such diversity may generate trajectories that do not necessarily align with human intent or expectations.

A key feature of our method is its dual utility: not only for deployment but also for collecting additional data to fine-tune the policy. For distribution shifts, our uncertainty metrics can identify states that require additional data, enabling targeted human supervision to improve the policy's performance. In the case of partial observability, humans can intervene to guide the robot in making critical decisions necessary to complete tasks. For action multi-modality, human guidance can help steer the robot into states where mode selection becomes unambiguous, allowing it to effectively execute one of its learned skills.

- We propose a simple yet effective method for building human-in-the-loop (HitL) policies by leveraging uncertainty estimation of a diffusion-based agent. Our approach eliminates the need for human-robot interactions during training and incurs minimal computational overhead during deployment.
- We validate our method across three key types of deployment challenges. Experimental results show that our approach effectively requests for human assistance only in critical states during deployment.
- We show that our critical state identification method can be utilized to collect targeted fine-tuning data, yielding significant performance improvements with smaller datasets compared to collecting full-trajectory demonstrations.
- To the best of our knowledge, we are the first to use uncertainty estimations with diffusion policies for human-in-the-loop deployment.

II. Related Work

A. Difficulties in policy deployment

While generalist real-world robot manipulation policies have made remarkable progress recently [8, 26], particularly in home environments [21, 37], several critical challenges persist in deploying these policies effectively. First, robots must handle significant data distribution shifts when deployed in novel environments. For instance, in the Amazon Picking Challenge, robots need to perform retrieval tasks across varied settings. Recent work SIMPLER [18] demonstrated that even minor changes, such as altering the robot arm's texture, can lead to a dramatic drop in success rate (over 20%). Second, real-world deployment faces incomplete observations due to environmental variables like lighting conditions and camera setups, requiring robots to overcome challenges including occlusion, clutter, and weak texture features [9]. Third, the inherent complexity of modeling multi-modal distributions in human demonstrations, combined with stochastic sampling and initialization procedures, presents significant challenges that have been extensively discussed in behavior cloning literature [7, 11, 24, 30].

Our work addresses these deployment challenges through a novel approach: strategically incorporating human assistance at critical moments. By identifying high-uncertainty states during deployment, our method enables timely human intervention to help the robot overcome distribution shifts, handle incomplete observations, and navigate complex multi-modal action spaces more effectively.

B. Human-in-the-loop policy

HitL approaches have been widely explored to enhance robot manipulation policies through various forms of human feedback, including interventions [24, 32], preferences [17], rankings [4], scalar-valued feedback [23], and human gaze [39]. Recent works like HIL-SERL [22] and Sirius [20] further demonstrate the benefits of human assistance - HIL-SERL achieves high performance in vision-based real-world RL. At the same time, Sirius optimizes behavioral cloning by incorporating human trust signals. The human-in-the-loop paradigm is also a well-recognized as a practically effective method in deploying self-driving cars. For instance, ZOOX designs user interfaces [25] for human operators to intervene their self-driving cars when they are stuck on the road.

However, these existing approaches primarily focus on incorporating human feedback during training without addressing when human assistance is most needed during deployment. They often require extensive human supervision throughout the process, which can be inefficient and impractical in real-world applications. In contrast, our work introduces an uncertainty-aware diffusion model that can actively identify critical moments requiring human expert intervention during deployment. This enables more efficient utilization of human expertise by requesting assistance only when the system's uncertainty is high, leading to a more practical and scalable human-in-the-loop framework.

C. Diffusion models for policy

Recent works have demonstrated the remarkable success of diffusion-based policies in robotics and decision-making tasks [2, 7, 27, 29, 33, 35, 38]. These policies excel at modeling complex behaviors and capturing multi-modal trajectory distributions when trained on high-quality demonstration data.

However, collecting perfect demonstration datasets is often impractical due to limitations in data collection and the presence of suboptimal demonstrations. To address this challenge, researchers have proposed various solutions. One line of work focuses on guiding the diffusion denoising process using external objectives, such as reward signals or goal conditioning [1, 5, 14, 19, 34]. Other approaches leverage techniques like Q-learning and weighted regression, either

through purely offline estimation [6, 36] or with online interactions [12, 15, 28].

Our work takes a fundamentally different approach by leveraging the distribution modeling capability of diffusion models. We observe that diffusion models' ability to capture the underlying data distribution can be utilized to quantify the uncertainty in action modes for each state. This unique perspective enables us to identify critical states with high uncertainty where human assistance would be most beneficial, leading to more targeted and efficient human-in-the-loop intervention.

III. Method

Our method is designed to determine when the agent should request expert assistance, ensuring optimal use of a limited number of such calls during deployment. Additionally, we aim to eliminate the need for expert intervention during the training phase. This means that the agent has no knowledge about the effect of an assistance except for the assumption that it would improve its task performance.

To achieve this, our method leverages the agent's internal uncertainty. Specifically, we utilize diffusion models as our policy class [7]. Diffusion policies offer two key advantages: (1) they demonstrate robust performance in imitation learning tasks, and (2) their generative process involves an iterative denoising mechanism, which provides insights into the agent's decision-making process. Their ability to effectively capture action multi-modality in human demonstration data contributes largely in their success in robot learning. In this section, we first introduce diffusion policies, then describe how our method utilizes its generative process to compute an uncertainty metric, and finally discuss how this metric can be applied to improve policy deployment performance.

A. Background: Diffusion Policy

Diffusion policies generate actions through an action-denoising process, leveraging denoising diffusion probabilistic models (DDPM). A DDPM models a continuous data distribution $p(x^0)$ as reversing a forward noising process from x^0 to x^K, which is defined as a Markovian chain with Gaussian transition. The reverse process is to predict the transition (i.e. noises added during each step) and map the data from x^K back to x^0. Sampling begins with a random input and iteratively refines it to produce a denoised output.

Specifically, the generative process of a diffusion policy $\pi(A|O)$ starts by sampling a random noise a_t^K and iteratively remove noises by:

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

To train a diffusion policy, we learn a score function ϵ_θ using a MSE loss:

$$\mathcal{L} = \mathcal{MSE}(\epsilon, \epsilon_\theta(o_t, a_t^k, k)) \tag{1}$$

Note that when we use a diffusion policy with task space control, it can be seen as partial forward models that estimate the end-effector pose in future time steps from current observations and current poses.

B. Human-in-the-loop Diffusion Policies

In this work, we aim to enhance the deployment performance of diffusion policies by incorporating human assistance. Our approach estimates an uncertainty metric for the policy, which is used during deployment to determine when human intervention is most beneficial. Importantly, the policy is trained using an offline dataset and does not rely on human assistance during the training phase.

To estimate the uncertainty of a diffusion-based agent, our method leverages the generative process inherent in the diffusion policy. As described in Section III-A, a diffusion policy generates actions by iteratively predicting the noise required to reconstruct the training data distribution. When operating in task-space control, where the action space represents end-effector poses, the predicted noise can be interpreted as a vector field pointing toward the target distribution in the training data.

Hence, in this work, we leverage this vector field to analyze whether a diffusion-based agent is confident about its generative target. In this work, we assume that our policy is operating on task space control and a diffusion policy outputs absolute end-effector (e.e.) poses and manipulator state. We also assume that the current e.e. pose is available as an input for action denoising. Our goal is to estimate an uncertainty metric $\texttt{Uncertainty}(o_t)$ where o_t is the observation at time step t.

To compute this metric, we first sample a set of points $A_{sampled}$ within a distance r from the current pose. For each sampled point, we feed the diffusion policy forward and predict the noise vectors required to sample actions:

$$v = \epsilon_\theta(o_t, a_{sampled}, 0) \tag{2}$$

These predicted noise vectors represent the directions toward the data distribution that the policy aims to recover. In this work, we use these denoising vectors to estimate uncertainty, defined as $\texttt{Uncertainty}(o_t) = f(V)$, where V represents the set of denoising vectors.

A critical characteristic of human demonstration is its multi-modality, which means that naive variance estimation of the vector field may fail to provide meaningful uncertainty information. To address this, we leverage Gaussian Mixture Models (GMMs) to capture the multi-modal nature of action generation. As detailed in Algorithm 1, our method first fits the collected denoising vectors with n GMMs, each using a different number of modes. We then select the best-fit GMM for uncertainty estimation:

$$D(v) = \frac{1}{k}\sum_{i,j} 1 - S_c(g_i, g_j)$$

where,

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

We also evaluate its variance as part of the uncertainty

Algorithm 1 HitL Policy Deployment

while rollout not done **do**
  Sample a set of points uniformly within the radius of r
  Feed forward the diffusion policy π to collect a set of vectors V
  Estimate uncertainty in this state using Eq.4
  if $D \geq D_{threshold}$ **then**
    Execute an action a_{human} from human input
    Save intervention data (o_t, a_{human}) to $\mathcal{D}_{int}$
  else
    Execute an action a_t from the policy $\pi(a_t|s_t)$
  end if
end while
if fine-tune **then**
  while fine-tuning not done **do**
    Sample a batch of data from $\mathcal{D}_{ft}$ and $\mathcal{D}_{train}$
    Optimize π using Eq. 1
  end while
end if

estimation:

$$\texttt{Var}_g(v) = \sum_n p(v_n)\texttt{Var}(v_n) \tag{3}$$

Putting them together, we can estimate the overall uncertainty as:

$$\texttt{Uncertainty}(o_t) = D(v) + \alpha\texttt{Var}_g(v), \tag{4}$$

where α is a constant.

This uncertainty estimation considers two aspects of our diffusion-based policy: 1. the number of modes the policy may generate and how diverged they are; 2. the internal variance of each mode. Here, we use negative cosine similarity to evaluate mode divergence.

C. Uncertainty-based human intervention and policy fine-tuning

With estimated uncertainty, during deployment, we can set a threshold to determine whether we are asking for human assistance. In this work, we consider three types of deployment issues that may cause uncertainty in the generative process:

- **Data distribution shift.** This is common for any learning system. Specifically in robot learning, this distribution shift can be caused by any data used in the robot system. For example, visual observation distribution shift can be caused by change of lighting condition. A special case for robotics is the change of dynamics that is caused by interacting with novel objects.
- **Partial observability.** This issue is present in almost all the robotic systems. The common approach to solve it can be redesigning sensors, adding sensors or changing sensing locations. However, in this work, we argue that it is impossible to have a sensor setup that provides full observations for all the tasks. Hence, the aforementioned solutions are limited by a specific task.

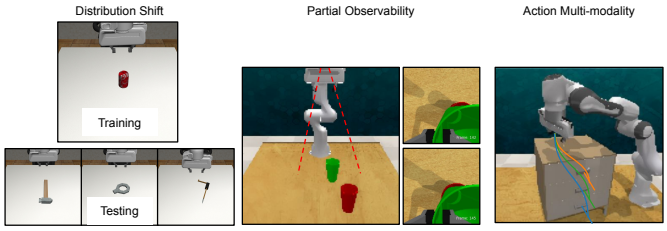


Fig. 2. **Simulated environments.** We considers three major problems during policy deployment. (a) Distribution shift; (b) Partial observability (c) Action multi-modality.

- **Mis-alignment between generated actions and human intents.** Human demonstrations are naturally multi-modal since humans are not good at reproducing the same trajectory. In some tasks, this diversity can produce action trajectories that achieve different goals. This is actually a data under-specification problem since the task description is not detailed enough to describe the expected behaviors.

Although all of these issues can be alleviated by human intervention, only data distribution shift and action multi-modality are suitable for policy fine-tuning to get better performance in a more autonomous manner. For partial observability, correct decision making is impossible without changing the available observation (e.g. use longer history of observations or change hardware to get better observation).

During policy execution, these problems may not be present in all states. In fact, many states are easy to make decisions. For example, moving the arm in free space is usually easy and does not require human attention to help the robot.

Our method uses the proposed uncertainty metric to determine whether to call for human assistance. In these states, a human can take control of the robot and tele-operate it until its uncertainty is low. Finally, 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 $\{O, A\}$ when a human operator is intervening with the robot and use this data set to fine-tune a diffusion policy. To avoid catastrophic forgetting [3], 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 $\mathcal{D}_{ft}$.

Putting all components together, the final pipeline contains three main steps: 1. train a diffusion policy; 2. deploy it with uncertainty estimation and employ human intervention; 3. if the problem can be resolved by fine-tuning, use the human intervention data to fine-tune the diffusion policy.

IV. Experiments

We validate our method on three types of deployment problems discussed in Section III-C, conducting experiments in both simulated environments and real-world robotic setups. In this study, we assume that humans are capable of tele-operating the robot to successfully complete tasks. Therefore, with sufficient human intervention and assistance, the task success

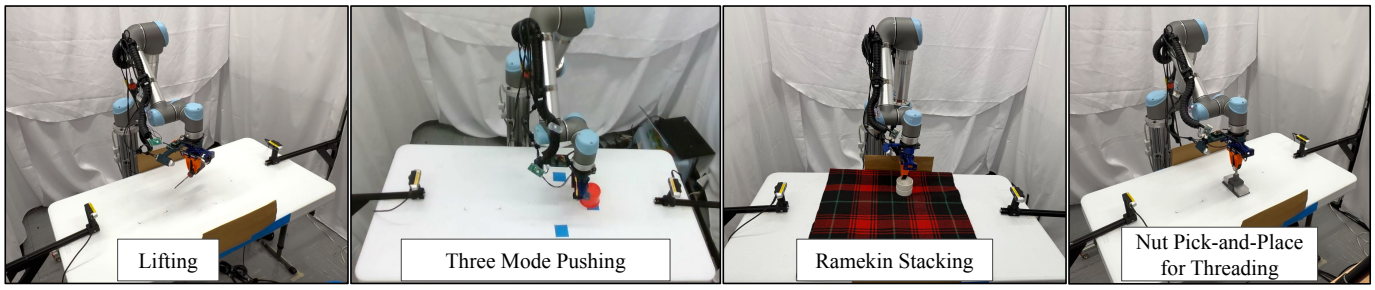


Fig. 3. Real robot experiments.

rate can theoretically reach 100%. However, a critical aspect of human-in-the-loop deployment is the efficiency of human intervention—the robot should request assistance conservatively, minimizing unnecessary interruptions.

To evaluate the effectiveness of our method, we focus on two key aspects across all experiments. First, we assess the efficiency of human-robot interaction by measuring the frequency and necessity of human interventions. Second, we evaluate the improvement in task performance achieved through human assistance and policy fine-tuning, quantifying the benefits of integrating human feedback into the system.

To benchmark our method, we compare it against two baseline approaches that incorporate uncertainty estimation. Unlike most offline learning methods, which lack uncertainty estimation, these baselines serve as meaningful points of comparison, as the absence of uncertainty estimation limits their adaptability to a human-in-the-loop framework. The baselines considered in this work are as follows:

- **Gaussian Processes (GP) Imitation Learning.** This baseline uses Gaussian Processes as the policy class for imitation learning, leveraging the natural uncertainty estimation provided by GP. Visual inputs are encoded into feature vectors using a pretrained CLIP visual encoder.
- **HULA-offline [31].** HULA is a human-in-the-loop policy learning method based on reinforcement learning. In this baseline, we augment the dataset by labeling rewards: each step receives a reward of 1 if it is the final step of the trajectory and 0 otherwise. While the original HULA method is designed for online RL, we adapt it to offline RL by implementing an offline variant using Conservative Q-Learning (CQL) [16], since only offline datasets are available during training.

A. Environments

In this section, we briefly summarize the environments that we test our method on in both simulated and real world setting. Details about each environment and statistics of collected data from them are included in the supplementary materials [ref].

Distribution shift: Lift-sim. we first consider the deployment problem of distribution shift. In this task, we ask the robot to grasp and lift objects in a table-top setting. During pretraining, demonstration data is collected using only a red cube (see Fig.4). For testing, we rollout the pretrained policy to grasp unseen objects (round nuts, hammers, and hooks) during training.

Partial observability: Cup Stacking. we then test our method on problems with partially observable environments. In this task, we ask the robot to grasp a green cup and place it inside a red cup stably. In this task, we use three views as our visual observation: a front view, a side view and a wrist view. Successful execution requires the robot to infer object alignment based on its observations. Misalignment can lead to unintended collisions, resulting in catastrophic failures, such as the red cup tipping over or becoming unstable. To introduce variability and train a robust policy, cup positions are randomized during data collection.

User intent misalignment: 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 approximately 33.3% of the data corresponding to each drawer. Importantly, the dataset does not specify which drawer is being opened in a given trajectory, introducing under-specification in the dataset.

B. Real robot experiments

Finally, we validate our method on real robot data collected using tele-operation. In this work, we use a tele-operation system to collect human demonstration data. The robot is controlled by a trakSTAR electromagnetic 6DoF pose tracker and a gripper control unit that provides continuous commands. To meet the need for real robot deployment, we use denoising diffusion implicit models (DDIM) that allow for high-frequency action generation. Since DDIM can be used with a DDPM, our method can be directly applied to a DDIM model. Unlike DDPM, our vector field sampling can be parallelized and batched feed-forward with the model. Hence, this additional computation does not add a big overhead during policy deployment. Details about hyper-parameters used with DDIM can be found in supplementary materials .

In this work, we evaluate our method on 4 real robot tasks (see Fig. 3). Similar as the simulated experiments, we show an example in each of the deployment problems.

Lift-real. This task extends the lift-sim setup to the real world, where the robot is trained to grasp a set of objects (Fig. 5) and tested on unseen objects to evaluate generalization. Although some objects from train and test set are visually

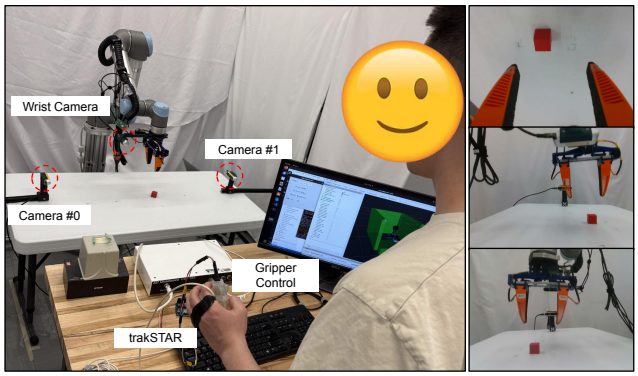


Fig. 4. Data collection pipeline.

Fig. 5. Objects used for the *Lift-real* task. The left and right images show training and testing objects respectively.

similar, the robot needs to learn different strategies to grasp them. For example, the cup we use for testing has a higher radius than the one in the training set, making it difficult to cage with a gripper. The robot needs to learn to grip the rim of the cup to achieve stable grasps.

3-Mode Pushing. In this task, the robot is required to push a cup to three designated locations on the table, marked with blue tape. The dataset used for training includes equal proportions of pushing trajectories for each of the three locations, with each mode a third of the full dataset. Notably, during training, the specific target location (mode) for the push is not explicitly provided to the robot.

Ramekin Stacking. In this task, the robot is required to pick up one ramekin and place it on top of the other. The success criterion for this task is achieving stable placement, where the top ramekin is horizontally aligned with the bottom one. This requires precise alignment between the two ramekins. As shown in Fig. 9, a key challenge arises during stacking because the bottom ramekin becomes occluded in the wrist camera view, making it difficult for the robot to achieve proper alignment for stable placement.

Nut Pick-and-Place. The final task evaluates our method on a pick-and-place scenario where the robot must pick up a nut and place it onto a lug. The success criterion for this task is achieving a stable placement of the nut, ensuring it is properly aligned for threading. This task demands high placement precision, which is particularly challenging to achieve using visual observations alone.

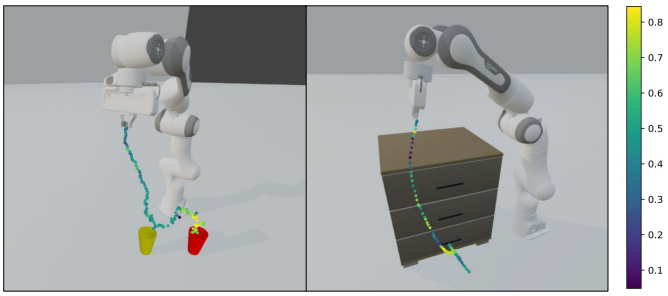
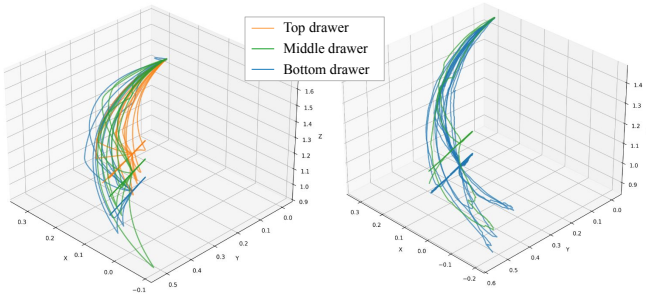


Fig. 6. Qualitative results for uncertainty estimation of the learned policy with the simulated task. The color of the trajectory represents the uncertainty estimation of the agent. Each dot represents a action prediction from the policy.



(a) Training data (b) Policy rollout

Fig. 7. Multi-modal demonstrations and learned modes from diffusion policy in the *3-Mode Pushing* task.

V. Results

A. Task performance

As a sanity check, we first evaluate the task performance of the diffusion policy on each task under the training data distribution. For the *Lift-sim* task, diffusion policy achieves a 100% success rate with the training object but fails completely (0% success rate) when tested on unseen objects.

In the *Pick and Insert Cup* task, the robot consistently picks up the green cup (100% grasp success rate) but fails to place it into the red cup due to alignment difficulties, resulting in an overall success rate of 0% without human assistance. We also note that this task is sensitive to observation selection—training with only side and front views causes the robot to fail at grasping the green cup.

For the *Open Drawer* task, the diffusion policy successfully learns to open a drawer with 100% success if the specific drawer is unspecified. Interestingly, despite the under-specified training (i.e., no conditioning on which drawer to open), the policy captures the multi-modalities of the training distribution. As shown in Fig. 7, during 100 rollouts, the robot opens the middle and bottom drawers in 15% and 85% of trials, respectively, but never opens the top drawer with random sampling.

TABLE I
AVERAGE # OF HUMAN ASSISTANCE STEPS FOR SIMULATED TASKS

	Lift-sim	Cup stacking	Open drawer
HULA-offline	56	54	22
Denoising Uncertainty (ours)	20	5	8
Full-trajectory length	80	147	115

TABLE II
HITL CONTROL EFFICIENCY FOR REAL ROBOT TASKS: AVERAGE # OF HUMAN ASSISTED STEPS DURING POLICY DEPLOYMENT

	Lift-real	Ramekin stacking	3-Mode Pushing	Nut PnP
Denoising Uncertainty (Ours)	7.2	6.8	6.5	8
Full-trajectory length	80	112	99	49

B. *Efficiency of human interaction*

We then evaluate human-in-the-loop deployment performance of these tasks. As mentioned in Section IV, the policy deployment can always achieve 100% success rate with human assistance. Hence, in this evaluation, it is important to look into the number of human assistance steps used to complete the task.

As shown in Table I, for all simulated tasks, our results show that the policy can achieve the task with small number of human steps. Compared to the baseline, our method can consistently complete the task with less human interventions. In all tasks, HULA-offline requires many steps to assist the robot because of its mis-estimation of uncertainty of the agent. For instance, for the open drawer task, it assign high but very similar uncertainty values to all states whose z-axis is lower than the middle drawer. This make threshold difficult – if threshold is too high, we cannot complete the task; if the threshold is too low, more human assistance is needed.

In contrast, our method can identify crucial states that allows for successful task completion using small number of human steps. For the *Lift-sim* task, the robot only seeks human assistance when its gripper is close to the object, allowing the human to pose the gripper to locations that leads to successful grasp. After the human assist the robot to grasp the object, it can lift it without any human intervention. For the *Pick and insert cup* task, our method identify states where the agent fails to align the two cups as high uncertainty. In a similar case, where the robot needs to pick up the green cup, since it has a full observation to complete the grasping, it has low uncertainty. This shows that our uncertainty estimation can capture how partial observation affect an agent's decision making. Finally, for the open drawer task, we shows that we allow users to choose one of the modes that is learned. The policy asks for assistance when it is reaching to one of the drawer. Once the human operator steers it to a certain height, the robot can autonomously complete the open drawer tasks. We also want to mention that the robot can stably open all the drawers stably with human assistance.

C. *Fine-tuning performance*

A key feature of our method is its ability to identify states where the policy exhibits high uncertainty. These states are particularly valuable as they highlight areas where the policy can benefit from additional data collection for fine-tuning. By addressing these high-uncertainty states, the policy can achieve improved autonomous performance and greater robustness in future deployments.

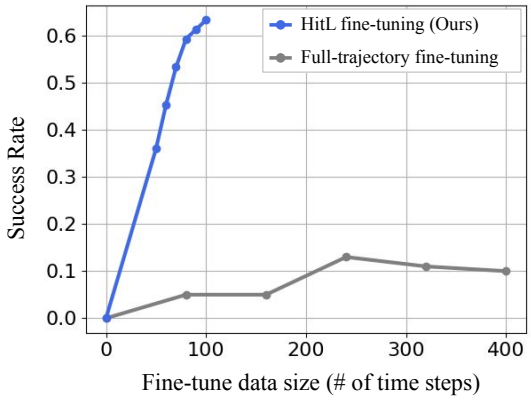


Fig. 8. Average success rate of fine-tuning the *Lift* task with different number of human demonstration samples.

Our experimental results demonstrate that leveraging our uncertainty metrics effectively reduces the amount of data required for fine-tuning while still achieving significant performance improvements. In this evaluation, we utilize human-assisted trajectories to fine-tune the diffusion policy. To mitigate the issue of catastrophic forgetting during fine-tuning, we adopt a balanced sampling strategy. Specifically, for each mini-batch, data is sampled equally from both the pretraining dataset and the fine-tuning dataset, with each comprising 50% of the batch. This approach ensures that the policy retains knowledge acquired during pretraining while incorporating new information from the fine-tuning process, thereby maintaining robust overall performance.

As shown in Fig. 8, our policy performance improves by 63.3% on average if we collect 160 time step of data whereas it only improves the policy by 28% success rate if we collect the full trajectory. This shows that our method can identify states that the policy needs more information. Compared to the baseline, our method also consistently achieves higher success rates with the same amount of fine-tuning data. Note that

TABLE III
FINE-TUNING PERFORMANCE OF THE *Lift-real* TASK.

	Train	Test
Zero-shot	1	0.16
HitL fine-tuning	1	0.63

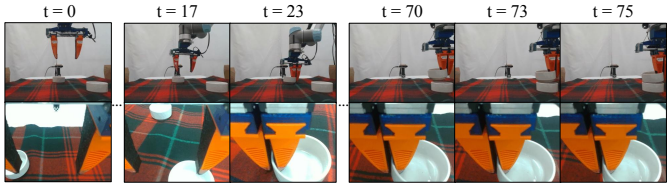


Fig. 9. Examples of observations for *Ramekin Stacking* in different stages. The top row shows image frames from one of the side cameras and the bottom row shows the view of the wrist camera.

the data used for each fine-tuning experiments are collected independently (i.e. the human-in-the-loop fine-tuning data set 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.

D. Evaluation on real robot experiments

Finally, our results on real robot experiments shows that our method can be used with policy trained with real world data. With human-assisted 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 during policy execution (see Table II). Since we are using action chunking during real robot deployment, we allow the human to control the robot four steps, the same as the diffusion policy. For all tasks, the robot asks for assistance for less than 6 times (each assistance with 3 human control steps). Details about the evaluations protocol (e.g. the number of rollouts for each evaluation) are included in supplementary materials.

Qualitatively, our method identify crucial states during policy execution. For example, in the *Lift-real* task, the robot ask for assistance when the gripper is close the the object. Using human-collected data, we can fine-tune the diffusion policy to improve 47% success rate on average. In the *3-Mode Pushing* task, the robot can autonomously reach to the other side of the object and transfer control to the human operator for posing the gripper to different relative poses to push to different regions. Another common case in this task is that the robot ask for assistance while it is in the center of 3 modes. As shown in Fig. 11, with policy steering, we can stably achieve different goals whereas policy random rollouts can only reach 2 out of 3 goals. In the *Ramekin Stacking* task, our method asks for assistance during stacking. As shown in Fig 9, the bottom ramekin is visually occluded, making alignment difficult.

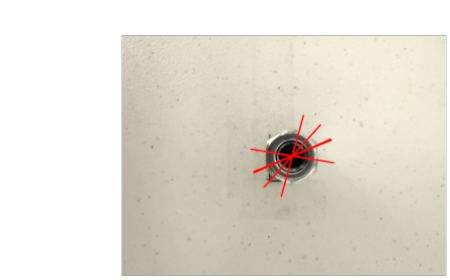


Fig. 10. Rotation of the gripper around z axis while grasping the nut with the same initial pose of the nut. The background nut image is used only for visualization, not observations.

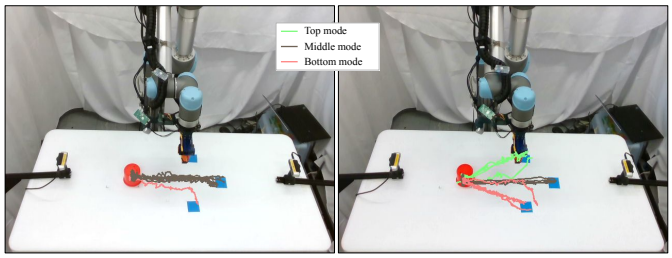


Fig. 11. Object trajectories visualization: policy random rollout vs policy steering.

Finally, in the *Nut Pick-and-Place* task, our method assigns high uncertainty to two critical stages of execution. The first stage occurs during the positioning phase for grasping, where the dataset contains diverse strategies for aligning the gripper with the nut's edge. Interestingly, when rolling out the policy multiple times with identical initial nut configurations, the robot exhibits varying rotations around the z-axis to grasp the nut (see Fig. 10). This highlights the ability of our uncertainty metrics to capture critical decision points during execution. The second stage of high uncertainty arises during the placement phase, 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 during this stage.

VI. ABLATION STUDY

TABLE IV
ABLATION STUDY: EFFECT OF SAMPLING RADIUS ON FINE-TUNING PERFORMANCE OF THE *Lift-sim* TASK.

Radius of sampling	0.01	0.03	0.05	0.1
# of fine-tuning steps (↓)	60.3	31.6	**20**	46.3
Success rate (↑)	0.46	0.55	**0.63**	0.53

In this section, we investigate how hyper-parameters affect the performance of our HitL agent. The choice of these hyper-parameters plays an important role of our uncertainty estimation, and hence can affect how the robot ask for human assistance.

A. Sampling Radii r

The radius parameter determines the neighborhood size for collecting denoising vectors used in uncertainty estimation. We

evaluate its impact in two experiments: fine-tuning performance on unseen objects for the *Lift-sim* task and human-robot interaction efficiency with the *Cup Stacking* task.

In HitL fine-tuning for *Lift-sim* (see Table IV), we test radii ranging from 0.01 to 0.1. A radius of 0.05 achieves the best balance between intervention steps and success rate, requiring only 20 interventions while improving the success rate by 0.63. This efficiency results from accurate uncertainty detection when the gripper approaches unseen targets but fails to grasp them. Smaller radii (0.01, 0.03) and larger radii (0.1) yield less precise estimations, leading to interventions that are either premature or delayed, reducing success rates despite more interventions.

In the cup pick-and-place task, where occlusion affects the final placement phase, an optimal radius (0.03 – 0.05) detects high variance during critical moments (steps 137 – 139), accurately identifying challenges caused by partial observability. Larger radii (0.2 – 0.5) shift high-variance detection earlier in the trajectory (steps 47 – step 29), introducing noise and obscuring uncertainty patterns.

Overall, a radius of 0.05 consistently achieves optimal performance by balancing local and global uncertainty estimations. Smaller radii miss key trajectory patterns, while larger radii incorporate irrelevant vectors, reducing the accuracy of uncertainty estimation in robotic manipulation tasks.

B. Scaling constant α in uncertainty calculation

The alpha parameter serves as a scaling factor in our uncertainty calculation, balancing two components: mode divergence and overall variance. Mode divergence captures directional differences between action modes using cosine similarities, while overall variance measures spread within each mode.

Directional differences typically provide stronger uncertainty signals, and small α values (0.01 – 0.1) emphasize mode divergence, effectively identifying critical occlusion phases (e.g., step 139). This highlights mode divergence as a reliable indicator of uncertain states.

As α increases (0.3 – 0.5), the overall variance term gains more weight, raising both maximum (0.93 – 1.13) and minimum (0.075 –0.101) variance values. However, this added emphasis on within-mode spread does not significantly enhance uncertainty detection, supporting the dominance of mode divergence as the more informative component.

The variance term remains essential for distinguishing states with single action modes, where mode divergence alone would yield identical scores. While α affects absolute uncertainty values, it has minimal impact on identifying critical steps (consistently around steps 137–139). This robustness demonstrates that our method effectively captures task-relevant uncertainties primarily through mode divergence, with α fine-tuning the signal.

VII. Limitations

Our work investigates methods for seeking human assistance in a non-interruptive manner. However, a key limitation of this approach lies in the lack of problem identification. This limitation restricts our method from transferring full control of the robot to human assistance when necessary. While human-in-the-loop policies present a promising direction for deploying robots, it is crucial to identify the root causes of deployment problems to enhance autonomous performance in the future. In this work, there is no mechanism for autonomously classifying deployment issues. Nonetheless, we empirically observe that analyzing uncertainty from different dimensions of the denoising vector can sometimes provide insights into the problem's source. For instance, high uncertainty in position compared to rotation may indicate distinct underlying challenges.

VIII. Conclusions

In this work, we propose a novel method that enables a robot actively and efficiently requests for humans' assistance during deployment. By utilizing an uncertainty-based metric, our method identifies situations where human intervention is most beneficial, thereby reducing unnecessary monitoring and intervention. Experimental results demonstrate the versatility of our method across various deployment scenarios, significantly improving policy performance and adaptability in real-world conditions. This work addresses one of their key challenges in human-in-the-loop robot deployment, minimizing human labor while maximizing robot autonomy and reliability. Additionally, our approach highlights the potential for using such interaction-driven methods to refine and fine-tune policies through targeted data collection.

For future work, we aim to further automate this process by exploring what types of information most effectively facilitate human-robot communication. Specifically, we will investigate how to design interpretable feedback that allows robots to convey their uncertainty and intent in a manner that is intuitive for human operators. Furthermore, we will study advanced control mechanisms that enable humans to seamlessly intervene and guide the robot when necessary. These efforts will help bridge the gap between fully autonomous systems and human-in-the-loop deployment, enabling more efficient and scalable solutions for real-world robotic applications.

References

[1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

[2] Lars Ankile, Anthony Simeonov, Idan Shenfeld, and Pulkit Agrawal. Juicer: Data-efficient imitation learning for robotic assembly. *arXiv preprint arXiv:2404.03729*, 2024.

[3] 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.

[4] 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.

[5] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024.

[6] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548*, 2022.

[7] 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.

[8] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[9] 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.

[10] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1etN1rtPB.

[11] 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.

[12] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[13] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, November 2024. ISSN 1558-2868. doi: 10.1145/3703155. URL http://dx.doi.org/10.1145/3703155.

[14] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[15] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] 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.

[17] Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021.

[18] 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.

[19] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

[20] 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.

[21] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[22] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning, 2024.

[23] 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.

[24] 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.

[25] Cade Metz, Jason Henry, Ben Laffin, Rebecca Lieberman,

and Yiwen Lu. How self-driving cars get help from humans hundreds of miles away. *New York Times*, 2024. URL https://www.nytimes.com/interactive/2024/09/03/technology/zoox-self-driving-cars-remote-control.html.

[26] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[27] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

[28] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. *arXiv preprint arXiv:2312.11752*, 2023.

[29] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[30] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[31] 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.

[32] 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.

[33] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[34] Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. *arXiv preprint arXiv:2309.06599*, 2023.

[35] Lirui Wang, Jialiang Zhao, Yilun Du, Edward H Adelson, and Russ Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024.

[36] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[37] Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib, Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. In *Conference on Robot Learning*, 2024.

[38] 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.

[39] Ruohan Zhang, Akanksha Saran, Bo Liu, Yifeng Zhu, Sihang Guo, Scott Niekum, Dana Ballard, and Mary Hayhoe. Human gaze assisted artificial intelligence: A review. In *IJCAI: Proceedings of the Conference*, volume 2020, page 4951. NIH Public Access, 2020.