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

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

***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, this approach is labor-intensive, requiring continuous human monitoring and intervention, which becomes impractical when deploying a large number of robots. To address this limitation, we propose an uncertainty-based metric for policy evaluation that actively seeks human assistance only when necessary, reducing reliance on constant human oversight. Our method leverages the generative process of diffusion policies and eliminates the need for human-robot interaction during training. Experimental results demonstrate that our approach effectively addresses various deployment challenges, enhancing policy performance during deployment. Additionally, we show that our method facilitates efficient data collection for fine-tuning diffusion policies, further improving their adaptability and effectiveness.**

I. Introduction

Recent advances in foundation models have shown significant potential for developing data-driven approaches to creating foundation behavior models (FBMs). The goal of an FBM is to generate action sequences for a wide range of tasks, ultimately achieving human-level dexterity. However, prior research on foundation models has revealed limitations: even large models trained on massive datasets can make errors in simple inference tasks due to hallucinations [12]. 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 prioritize not only task performance but also performance improvement through human assistance. For instance, Singi and He et al. [29] propose using return variances during reinforcement learning (RL) as a metric to determine when to request human intervention. However, these approaches often exhibit instability and are highly sensitive to the scale of the trained model, as they rely on learning a value function from reward signals, which can vary significantly.

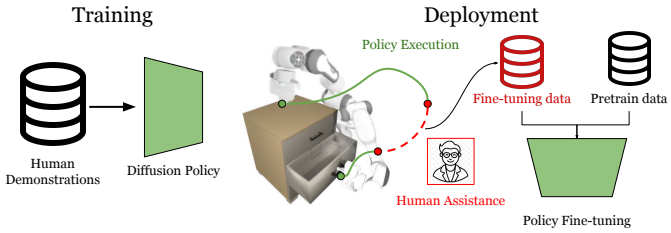

Fig. 1. **Human-in-the-loop policy** Our methods aims to develop human-in-the-loop policies that actively seek for human assistance. During deployment, the robot can ask for human assistance if its uncertainty estimation is high (shown in red). In these states, an operator can take control of the robot until its uncertainty is low. Finally, we can also save the human-operated data and use it to fine-tune a policy to obtain better autonomous execution performance in the future.

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 data under-specification.

While distribution shifts and partial observability are well-known challenges in robot learning [16], data under-specification 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 methods to capture the uncertainty of a diffusion-based agent. Using this uncertainty metric, our method effectively identify key states in needs for human assistance during policy execution.
- We demonstrate our method in three important types of deployment issues. Our experimental results shows that our method can find states with deployment issues and improve deployment performance with the help of human operators.
- We show that our key state identification method can be used for collecting fine-tuning data to improve policy performance, resulting in higher performance gain with small amount of data than collecting full-trajectory demonstration.

II. Related Work

A. Policy deployment issues

While generalist real-world robot manipulation policies have made remarkable progress recently [8, 24], particularly in home environments [19, 35], 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 [16] 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, 10, 22, 28].

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 [22, 30], preferences [15], rankings [4], scalar-valued feedback [21], and human gaze [37]. Recent works like HIL-SERL [20] and Sirius [18] 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 [23] 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, 25, 27, 31, 33, 36]. 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, 13, 17, 32]. Other approaches leverage techniques like Q-learning and weighted regression, either through purely offline estimation [6, 34] or with online interactions [11, 14, 26].

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 expect for the assumption that it would improve its task performance.

To achieve this, our method leverage an agent's internal uncertainty. Specifically, in this work, we use diffusion models as our policy class [7]. The advantage of diffusion policy is two fold: 1. it demonstrates its robust performance for imitation learning; 2. its generative process is a iterative denoising process, which provides information about an agent's decision making. Its success in robot learning largely comes from its ability to capture action multi-modality in human demonstration data. In this section, we will first introduce diffusion policy, then talk about how our method utilize the generative process to maintain an uncertainty metric, and finally discuss how this metric can be use in policy deployment.

A. Background: Diffusion Policy

Diffusion policy generates actions through an action-denoising process, leveraging denoising diffusion probabilistic models (DDPM). A DDPM models a continuous-valued data distribution by reversing a forward noising process, where Gaussian noise is progressively added to the data over several iterations. The reverse process is parameterized by a neural network that predicts the noise added during each step, effectively mapping 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:

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

Note that when we use 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 Policy

In this work, we aim to enhance the deployment performance of a diffusion policy by incorporating human assistance. Our approach estimates an uncertainty metric for the policy, which can be utilized during deployment to determine when human intervention is beneficial. Importantly, the policy is trained using an offline dataset and does not have access to human assistance during the training phase.

To estimate the uncertainty of a diffusion-based agent, our method leverages the generative process underlying 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 using 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 embedded 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 e.e. poses. 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 the time step t.

We first sample a set of points whose distances are within r. For each of these points, we feed-forward the diffusion policy and predict the noises needed to sample actions. Note that the predicted noise represents the direction of the data distribution that our policy need to recover. Hence, in this work, we use these denoising vector for uncertainty estimation: $\texttt{Uncertainty}(o_t) = f(V)$.

A key characteristic of human demonstration is multi-modality. This means that a naive variance estimation of the vector field may not gives us informative information of the agent. Building on this understanding, our work leverage Gaussian Mixture Models (GMM) to capture the multi-modality of action generation. As shown in Algorithm 1, our method first fits the collected denoising vectors with n GMM using different number of modes. Then, we use the best-fit GMM for uncertainty estimation. To capture the multi-modality behavior of human demonstrations, we consider two aspects of the denoising directions: 1. divergence of modes and 2. mode variations. We first evaluate the cosine-similarity between each pairs of modes:

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

where,

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

For each mode, we also evaluate its variances:

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

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

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

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 cosine similarity to evaluate mode similarity and the maximum of this value is 1, representing two vectors are perfectly aligned with each other.

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:

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 to collect a set of vectors V
  Estimate uncertainty D in this state using Eq.
  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 $\mathcal{D}_{ft}$.
    Update policy parameters with data.
  end while
end if

- **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 robotics system. The common approach to solve it can be redesigning sensors, adding sensors or change sensing locations. However, in this work, we argue that it is impossible to have a sensing set up that provides full observations for all the tasks. Hence, the aforementioned solutions is limited by a specific task.
- **Action multi-modality.** 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 achieved 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's attention to help the robot.

Our method uses the proposed uncertainty metric to determine whether call for human assistance. In these state, 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.

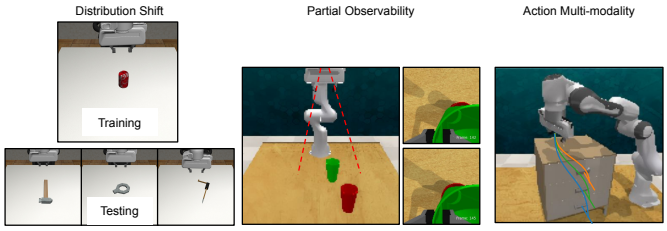

Fig. 2. **Simulated environments.** We considers three major issues during policy deployment. (a) Distribution shift; (b) Partial observability (c) Action multi-modality.

To fine-tune a policy, we save the observation and action pairs $\{O, A\}$ when a human operator is intervening the robot. 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 pretrain dataset $\mathcal{D}_{train}$. For each mini-batch, we ensure 50% are from D_{ft}.

Putting all components together, the final pipeline contains three main steps: 1. training 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.

D. Implementation details

In this work, we use a transformer-based diffusion policy (DP-T) [7]. For the task we are testing, we find that a transformer-based diffusion policy provide us stable task performance. We will discuss more in details in Section about the choice of architecture.

IV. Experiments

We validate our method with three types of deployment issues discussed in Section III-C in both simulated environments and real robot setup. To understand the effectiveness of our method, we look into two aspects of evaluation for all our experiment: 1. efficiency of human-robot interaction; 2. task performance improvement with human assistance and policy fine-tuning. In this section, we first discuss our experiment setup, then present and analyze the results.

A. Tasks descriptions

We test our method on three simulated environments. For all of them, we first pretrain a diffusion policy with collected demonstrations, evaluate its task performance with and without human assistance, and evaluate a fine-tuned policy performance. Here, we list the simulated environment used:

1) Lift (distribution shift): In this task, we ask the robot the grasp and lift objects. For pretraining, we collect demonstration data on only a cube object (see Fig.3). During testing, we ask the robot to lift three different types of objects (round nuts, hammers, hooks). This testing evaluates the generalization ability of a diffusion policy on unseen dynamics during training.

For observations, we use the side view and wrist view of the scene, as well as e.e. pose of the robot. The training dataset consists of 200 trajectories and 9666 steps.

2) Pick and insert cup (partial observation): In this task, we ask the robot to grasp the green cup in the scene and place it to the red cup. This task requires the robot to infer alignment between two objects from observations. In this task, we uses three views for policy learning: front view, side view and wrist view. The cup locations is randomized to create variations for training a robust policy. The training dataset have n trajectories.

3) Open drawer (action multi-modality): In this task, we ask the robot to open one of the three drawers in the scene. We collect data that open different drawers and each of them have about 30% of the dataset. For observations, we do not specify which drawer this trajectory open. We can see it as a dataset with three action modes. This dataset consists of n trajectories.

For each task, after we train a diffusion policy, we evaluate their performance by rolling out the closed-loop policy k times and record its success rate. In all tasks, the human help the robot by directly controlling it. For the simulated environment, the human operators control the robot using keyboard control. n actions are available by the human.

For each task, we first train a diffusion policy and then evaluate its performance by executing the closed-loop policy k times and recording its success rate. In all tasks, human operators assist the robot by directly intervening and taking control when necessary. In the simulated environment, the human operators use keyboard inputs to control the robot, with n discrete actions available for intervention.

B. Baselines

In this work, we compare our method with two baselines:

- **Gaussian MLP policy** in this baseline, we train an multi-layer-perception (MLP) policy to output the mean and covariance of a Gaussian distribution, which is used to sample actions. To train this policy, we use maximum likelihood estimation as our loss function to maximize the probabilities of training data.
- **HULA-offline [29]** in this baseline, we augment our dataset to have rewards. For each step, we label its reward as 1 if it is the last step of the trajectory and 0 for other steps. The original method bases on online learning RL. In this work, we train it with an offline dataset.

C. Evaluation

In this work, to evaluate the effectiveness of our method, we measure the success rate of the policy deployment with assistance from human. For each task, we roll out the policy in an environment for k times and record its success rate. It is also important to understand whether our method can reduce human operators' workload. Hence, the number of human intervention is used for measuring human intervention.

D. Real robot experiments

Finally, we show that our method can be used in a real world set up by learning diffusion model with real robot data. 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. We collect 50 trajectories for each task to learn a diffusion policy. Similarly with the simulated experiments, we evaluate its zero-shot performance on the real robot, then deploy our human-in-the-loop method to evaluate its performance.

To meet the need for real robot deployment, we use denoising diffusion implicit models (DDIM) that allows for high-frequency action generation. Since DDIM can be use with a DDPM, our method can be directly applied to a DDIM model. Unlike DDPM, our vector field sampling can be parallelized and batched feedforward with the model. Hence, this additional computation does not add a big overhead during policy deployment.

V. Results

A. Task performance

As a sanity check, we first evaluate the task performance of diffusion policy on each aforementioned task with training data distribution.

For the *Lift* task, we verify that diffusion policy can produce a 100% success rate policy with the training object. However, when we use it to lift unseen objects, its performance drops to 0%. Here, we also show some example of how diffusion policy fails on unseen objects. As shown in Fig, the robot fails to make decision on when to close the gripper.

For *Pick and insert cup*, the robot successfully picks up the green cup but fails to place it into the red cup. Hence, the success rate of this task without human assistance is 0%. We also want note that this task is sensitive to observation selection. As shown in Fig, if we train this task with only the side view and front view, the robot fails this task in the first step since it cannot align its end-effector with the green cup for grasping.

For *Open drawer*, the robot successfully learns to open one of the drawers with 100% success rate if we do not specify which drawer to open. Interestingly, although the diffusion policy is learned in an under-specified manner (i.e. we do not condition the policy with which drawer to open), diffusion policy can still capture the multi-modalities of the training distribution. As shown in Fig, we use this policy to roll out k times in the environment and it generate trajectories the open different drawers with stochastic sampling. However, if we want the robot to open one of the drawer, there is no way in diffusion policy to choose how to generate the trajectories. In this case, the success rate of opening the middle and lower drawer (i.e. selecting one of the mode) are only about 15% and 85% of 20 rollouts.

B. Efficiency of human interaction

We then evaluate our method with human intervention. As mentioned in Section, if the uncertainty estimation is higher than the threshold we choose, our robot hands control to a human operator. With human intervention, we can achieve 100% success rate of all three tasks.

TABLE I
QUANTITATIVE RESULTS OF FINE-TUNING THE LIFT TASK ON UNSEEN OBJECTS

# of Finetune Trajectories	Round Nut (%)	Hammer (%)	Hook (%)	Average (%)	Fine-tune Data Amount
0	0.00	0.00	0.00	0.00	0
High Uncertainty State data collection Fine-tuning					
5	2.00	40.00	66.00	36.00	100
6	24.00	36.00	76.00	45.33	120
7	26.00	58.00	76.00	53.33	140
8	50.00	58.00	82.00	63.33	160
9	56.00	46.00	82.00	61.33	180
10	60.00	50.00	80.00	63.33	200
Full-Trajectory data collection Fine-tuning					
1	0.00	10.00	0.00	3.33	80
2	0.00	10.00	0.00	3.33	160
3	46.00	22.00	4.00	24.00	240
4	44.00	18.00	4.00	22.00	320
5	64.00	12.00	8.00	28.00	400

TABLE II
ABLATION STUDY: EFFECT OF SAMPLING RADIUS ON FINE-TUNING PERFORMANCE OF THE *Lift* 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

For the *Lift* task, the human control the robot when it is close to the object. It shows that our uncertainty metrics can capture where it has trouble during execution. 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. Here, after human intervention, a robot can always open a specific drawer with 100% success rate.

Although we can always complete the task with human intervention, it worth looking into how many human-robot interactions are needed to achieved this performance. Our experiment results show that the robot asks for assistance for on average 20, 20 and 20 time steps out of 80, 80, 80 steps for a full trajectory. This means that the human operator only need to control the robot behaviors in about 25% of the time steps.

C. Fine-tuning performance

A key feature of our method is using the uncertainty metrics to collect data for fine-tuning. Instead of having human operators to control a robot to do the full task, we propose to control the robot only in interesting states to reduce workload of a human operator.

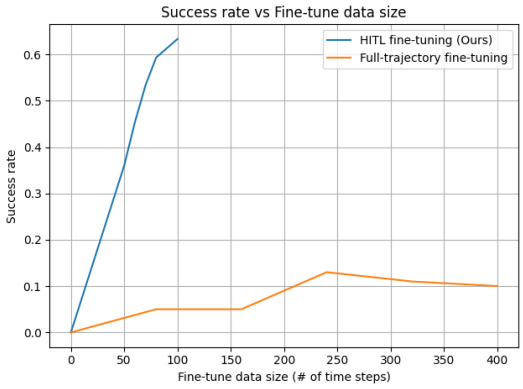

Fig. 3. Average success rate of fine-tuning the Lift task with different number of human demonstration samples.

Our experimental results show that using our uncertainty metrics can reduce the steps that a human need to control but still achieve high performance improvement. As shown in Table, 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.

In this work, we only fine-tune on tasks with distribution shift since fine-tuning is not a general solution for partial observability. Action multi-modality is both a feature and a problem. Hence, we do not fine-tune on this problem either to maintain multi-modality. Instead, during deployment, we uses our method to choose one the modes for execution.

D. Key design decisions

Uncertainty estimation Here, we discuss the key considerations for the choice of uncertainty estimation. A most straightforward way to estimate the uncertainty using the denoising vector is compute its variance. However, we find

TABLE III
EFFECT OF RADIUS AND ALPHA ON UNCERTAINTY CALCULATION AND HUMAN ASSISTANCE STEPS IN CUP PICK AND PLACE TASK

Parameter	Value	Max Variance Value	Min Variance Value	Max Variance Step	Min Variance Step	Human Steps	Whole Steps
Radius (5)	0.03	0.8442	0.0484	137	88	7	146
	0.05	0.7435	0.0485	139	88	5	145
	0.1	0.6346	0.2729	106	88	10	118
	0.2	0.5341	0.2469	47	88	81	123
	0.5	0.6416	0.4090	29	88	96	127
Alpha (5)	0.01	0.6600	0.0365	139	88	5	145
	0.05	0.6971	0.0418	139	88	5	145
	0.1	0.7435	0.0485	139	88	5	145
	0.3	0.9364	0.0750	137	88	7	146
	0.5	1.1346	0.1016	137	88	7	146

that a variance is not a good choice empirically since it does not capture the multi-modality of the actions.

As shown in Fig, the action directions in this state mainly falls into two modes. If we assumes it is unimodal, its variance is lower than many states. However, if we use GMM to capture its multi-modality, its uncertainty is higher.

VI. ABLATION STUDY

In this section, we investigate how hyper-parameters affect the performance of our human-in-the-loop 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.

Sampling Radii The radius parameter determines the size of the neighborhood when collecting denoising vectors for uncertainty estimation. We conduct two sets of experiments to evaluate its impact: one for zero-shot transfer learning with fine-tuning on unseen objects, and another for direct human assistance in a cup pick-and-place task with partial observability.

As shown in Table II, we evaluate the effect of different sampling radii (0.01 to 0.1) on the transfer learning performance. With a radius of 0.05, we achieve the optimal balance between intervention steps and success rate improvement. Specifically, this setting requires only 20 intervention steps while achieving a 0.63 success rate improvement. This efficiency stems from accurate uncertainty detection at critical moments when the gripper is close to the unseen target but fails to properly grasp it. At this radius, the collected expert demonstrations are more focused on teaching the policy the missing knowledge about grasping novel objects. Both smaller radii (0.01, 0.03) and larger radii (0.1) lead to less accurate uncertainty estimation, resulting in premature or delayed interventions. This noise in prediction causes the collected demonstrations to be less targeted at the policy's key knowledge gaps, leading to lower success rates despite more intervention steps.

Table III further validates these findings in a partially observable cup pick-and-place task. The task involves picking up a green cup and placing it into a red target cup, where occlusion naturally occurs in the final placement phase. With optimal radius values (0.03-0.05), our method consistently detects high variance near the end of the trajectory (steps 137-139), precisely when the green cup occludes the target red cup in the wrist camera view. This indicates our method successfully identifies the most challenging moment of the task where partial observability significantly impacts performance. As the radius increases to 0.2-0.5, the maximum variance steps shift earlier (47-29), suggesting that larger neighborhoods introduce noise that obscures the true uncertainty patterns arising from partial observability.

A radius of 0.05 achieves optimal performance across both experiments. Too small a radius restricts the neighborhood of collected vectors, leading to overly local estimations that miss important trajectory patterns. Conversely, larger radii incorporate irrelevant vectors from distant states, introducing noise that obscures genuine uncertainty patterns. This demonstrates how the sampling radius significantly impacts the accuracy of uncertainty estimation in robotic manipulation tasks.

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

Since directional differences between modes often provide stronger signals about action uncertainty, we use alpha to balance these two components. As shown in Table III, with small alpha values (0.01-0.1), the uncertainty calculation is dominated by mode divergence, effectively identifying the critical occlusion phase at step 139. This demonstrates that angular differences between action modes are indeed reliable indicators of uncertain states.

As alpha increases to 0.3-0.5, the overall variance term gains more weight, leading to higher maximum variance values (0.93-1.13) and elevated minimum variance (0.075-0.101). However, this increased emphasis on within-mode spread does not significantly improve uncertainty detection, supporting our hypothesis that mode divergence is the more informative component.

Notably, the variance term remains necessary even with

small alpha values, as it helps differentiate between states with single action modes where mode divergence alone would yield identical uncertainty scores. While the value of alpha affects the absolute uncertainty values, it has minimal impact on identifying the critical steps (consistently around 137-139) that require attention. This robustness suggests that our uncertainty estimation method effectively captures task-relevant uncertainties primarily through mode divergence, with the scaling factor alpha playing a secondary role in fine-tuning the uncertainty signal.

VII. Conclusion

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,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, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho,

Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. 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] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.