OOPS: Out-of-Distribution Policy Summarization

Harrison Huang, Yao Rong, Peizhu Qian, Vaibhav Unhelkar

***Abstract*— Robots are increasingly deployed across diverse real-world domains. As they assume critical roles, it becomes essential for humans to understand how they sense, reason, and act to ensure safe and responsible use. Recent advances in Explainable AI (XAI) have sought to support this understanding through the explanation-by-example paradigm. Specifically, policy summarization has emerged as a promising solution, generating summaries composed of representative examples of robot behavior for users. However, existing methods select examples only from the robot's training set (i.e., in-distribution scenarios). As such, they fail to capture a variety of scenarios that the robot would encounter in practice but are absent from its training set (i.e., out-of-distribution scenarios). We argue that holistic explanations, containing both in-distribution and out-of-distribution examples, are crucial for fostering human understanding of robot behavior. To address this gap, we introduce Out-of-Distribution Policy Summarization (OOPS), an explanation-by-example approach designed to jointly teach users how robots perceive, reason, and act in novel scenarios.**

I. Introduction

Robots are increasingly assisting human users, including at workplaces, hospitals, and disaster zones [1]–[4]. Across these domains, *humans' ability to anticipate robot behavior is crucial for safe and effective human-robot interaction (HRI)* [5]. To predict robot behavior, humans naturally form a Theory of Mind (ToM) about a robot based on their prior knowledge and interactions [6], [7]. However, without proper interventions, humans' acquisition of an accurate ToM can be slow, thereby leading to task failures and, in the worst case, catastrophic accidents resulting from poor HRI.

Fortunately, explanations can help. Research has shown that *users who are provided with explanations of a robot's decision-making can better predict its behavior* [8], [9]. Consequently, several XAI (eXplainable AI) methods have been designed to explain machine learning models, including those used for perception by robots, such as image classification and object detection algorithms [6], [7]. More relevant to our focus on robot behavior, policy summarization has emerged as a promising XAI solution [10]–[15]. Following the explanation-by-example paradigm, *policy summarization algorithms explain a robot's overall behavior (mathematically, its policy function) by creating summaries comprised of salient examples (demonstrations) of its behavior*.

Research Gaps: *Existing policy summarization methods, however, are limited in the content of summaries that they generate*. First, most assume the robot has complete observability of the task state, thereby omitting details on its perception [13], [16]. However, robot perception is rarely perfect; robots utilize perception models that map sensor observations to the robot's internal representations or actions. These perception models are trained on datasets of input-output pairs. Second, the few policy summarization works [17] that consider a robot's perceptual limitations are restricted to generating summaries based on the training dataset (henceforth, referred to as in-distribution or ID examples). As such, the *existing policy summarization algorithms fail to capture a variety of scenarios that the robot would encounter in practice but are absent from its training set* (henceforth, referred to out-of-distribution or OOD examples).

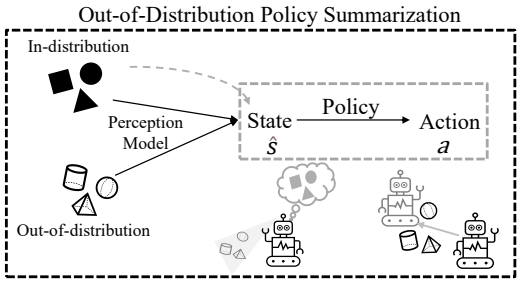


Fig. 1. Inspired by prior policy summarization methods, OOPS explains the robot's overall behavior through salient examples. However, unlike prior methods, OOPS is capable of generating out-of-distribution examples, thereby producing summaries that consider a richer variety of scenarios than those available in the robot's training set.

Summary of Contributions: We argue that *holistic summaries, containing both in-distribution and out-of-distribution examples, are crucial for fostering human understanding of robot behavior*. To enable generation of such policy summaries, our work makes the following contributions:

1) In Sec. III-D, accounting for robot's partial observability of the task state, we provide **a more general formulation of the policy summarization problem** that considers both robot perception and decision-making.
2) In Sec. IV, to solve this problem, we provide **an algorithm OOPS: Out-of-Distribution Policy Summarization** that generates policy summaries comprising of both in-distribution and out-of-distribution scenarios (Fig. 1).
3) In Sec. V, through **two human subject studies** evaluating the impact of robot policy summaries, we confirm that out-of-distribution examples are indeed critical to human understanding of robot behavior and can be effectively generated using OOPS.

The paper concludes with open problems and future directions on policy summarization, an important topic for safe and responsible human-robot interaction.

Huang and Unhelkar are affiliated with Rice University, Houston TX, USA (emails: hhuang@rice.edu, unhelkar@rice.edu). Rong is affiliated with the Leibniz University Hanover, Germany. Qian is affiliated with the University of Houston, USA. This work was supported in part by the NSF under award 2205454 and Rice University's Ken Kennedy Institute.

II. Related Work

Before describing our contributions, we provide a brief overview of methods for explaining robot behavior [8], [9], categorizing them based on their focus: perception or policies.

A. Explaining Robot Perception

A core challenge in XAI for robotics is making a robot's *perception* explainable for end users. In HRI applications, robots commonly rely on computer vision models to perceive humans and their environment. XAI methods can explain *how visual inputs influence output* of these perception models through feature attribution or concept-based reasoning [18]–[22]. Feature attribution methods, such as Grad-CAM, highlight image regions most influential to model prediction [20] and have been used for explaining robot perception [23], [24]. Concept-based explanation approaches link a model decision to human-interpretable concepts [21], [22]. These methods offer *local* explanations, explaining a specific (input, output) instance of the perception model. To help users understand the model's overall behavior, *explanation-by-example* paradigm builds on local explanations and offers salient examples of model's (input, output, local explanation)-tuples [25], [26]. While related, aforementioned XAI methods focus exclusively on explaining perception and do not consider robot policies. In contrast, *this paper jointly considers perception and decision-making, integrating explanations of perception models with frameworks for explaining robot policies.*

B. Explaining Robot Decision-Making

It is imperative that humans understand robot behavior, including its perception and decision-making, for safe and effective HRI [5]–[9]. Policy summarization methods have emerged as a practical mechanism for addressing this need [10]–[13]. These methods explain a robot's policy by generating summaries composed of salient demonstrations of its behavior. To generate summaries, some methods exploit information internal to the policy (e.g., value functions or Q-values) to identify salient states and actions for explanation [14], [15]. A complementary line of work explicitly models how humans infer robot intent using Theory-of-Mind formalisms, and then optimizes for demonstrations that are most informative under the user's inference model [11]–[13]. Evaluations with humans confirm the utility of these methods in enhancing user understanding and prediction of robot behavior [11]–[13]. Despite these advances, most existing policy summarization methods are limited to explaining state-to-action policies, ignoring robot perception. In contrast, *the proposed method OOPS also considers robot perception and explains observation-to-action policies.*

In summary, robot behavior depends not only on its state-to-action policy but also its observation-to-state perception [27]. This motivates XAI approaches that couple *policy* summaries with *perception* explanations, especially for OOD situations where misperception can induce downstream behavioral deviations. However, most existing methods for explaining robot behavior focus either on perception or policies, but not both. The proposed method OOPS addresses this gap.

III. Problem Formulation

This section describes mathematical models for the robot's task and behavior, followed by a formal description of the OOD policy summarization problem.

A. Task Model

Similar to existing work on policy summarization, we consider problem settings where the robot performs sequential tasks that can be described using a Markov Decision Process (MDP) [28]. MDPs are a general model of sequential decision-making under uncertainty, prevalent in robotics and embodied AI. Mathematically, an MDP is described by the tuple $\mathcal{M} \doteq (\mathcal{S}, \mathcal{A}, T, R, \gamma, h)$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the set of states s and actions a, respectively. $T(s'|s,a)$ is the transition function, capturing the probability of the impact of the robot's actions on the state transitions. $R(s,a)$ is the reward function, signifying the immediate reward. γ is a discount factor, and h is the task horizon. The MDP objective is to maximize the expected cumulative return, $G \doteq \mathbb{E}\left[\sum_{t=0}^{h} \gamma^t R(s_t, a_t); \pi\right]$, by selecting the optimal decision-making policy $\pi(a|s)$.

B. Robot Model

In practice, the robot often does not have complete knowledge of the task model, e.g., the MDP transition or reward function. Consequently, its behavior in practice might not be optimal. State-of-the-art policy summarization methods, therefore do not make assumptions about how the robot's autonomous behavior is programmed (e.g., it can be learned, planned, or even rule-based), nor do they assume that it is perfect [13], [16]. *However, existing methods typically assume that robot has full observability of the task state and focus on state-dependent policies of the type* $\pi(a|s)$.

In this work, we relax this strong assumption by allowing for partial observability of the task state. Specifically, we model that the robot observes its environment through sensors that provide observation $z \in \mathcal{Z}$, and then uses an *observation-dependent policy* that maps these observations to actions. Multiple methods exist for learning policy mappings from observations to actions [29]–[31]. A common approach involves a modular policy, where the robot first maps its observations to the most probable estimate of the task state, and then applies a state-dependent policy that maps this internal estimate to an action. We focus on this modular policy and discuss the generalizability of our contributions to other representations in the Conclusion section.

Mathematically, we model the robot's behavior through a modular observation-dependent policy $\pi : \mathcal{Z} \times \mathcal{A} \to [0,1]$,

$$\pi(a|z) \doteq g \circ f(z) \tag{1}$$

represented as a composition of robot's perception $f(z) = \hat{s}$ and decision-making $g(a|\hat{s})$ models. The perception model $f : \mathcal{Z} \to \mathcal{S}$ is typically a machine learning model, such as an object detection model [32], fine-tuned to estimate the task state. We denote the dataset used to train this model as $\mathcal{D}$, composed of model input-output pairs (z, s). Observations belonging to this dataset are denoted as *in distribution* (ID), while the remaining are considered *out of distribution* (OOD).

While the robot may encounter a large set of observations during task execution, $z \in \mathcal{Z}$, it is trained using a subset of this data, $\mathcal{D} \subset \mathcal{Z}$. The decision-making model $g : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ can be developed through a variety of algorithms [28], [33]. We do not make assumptions regarding the mechanism for computation of g or its optimality.

C. Limitations of Prior Work

Existing policy summarization methods summarize a robot's overall behavior, as defined by its *state-dependent* policy function, using n salient examples [10]–[15]. These methods, however, only consider in-distribution (ID) observations and policy summaries. Each example is a demonstration of robot trajectory, $\tau_i(s_i)$, obtained by executing the robot's policy $\pi(\cdot)$ starting from state s_i. Optionally, these examples can be augmented with local explanations e_i of the demonstrated behavior, such as reward information [13]. For these works, a policy summary of size n can be represented as a set of n states: $\sigma_s = \{s_1, \cdots, s_n\}$. The problem of (in-distribution) policy summarization is thus one of selecting the subset $\sigma_s \subset \mathcal{S}$ where demonstrations of robot behavior most effectively summarize its state-dependent policy.

While these methods have found success, they are limited by assumptions inherent in their problem formulation. First, they represent robot behavior using a state-dependent policy, thereby assuming that the robot has full observability of the task state. Consequently, they search for the policy summaries σ within the state space, assuming $\mathcal{Z} = \mathcal{S}$ comprehensively covers possible robot stimuli and resulting behavior. However, in practice, robots must make decisions based on observations. Thus, policy summaries must account for robot behavior across the broader range of stimuli encountered in practice, mathematically represented by its observation space $\mathcal{Z} \neq \mathcal{S}$. This work addresses this more generalized problem of policy summarization, which considers observation-dependent policies. The problem addressed in prior works, which we refer to as in-distribution policy summarization, is a special case of the proposed problem that additionally assumes f perfectly estimates the state from observations.

D. Problem Statement

Summaries of robot policies must summarize both robot's perception (e.g., *Which objects it perceives in the current scene?*) and action (e.g., *Which object it will pick next?*). As such, for observation-dependent policies, summaries should include not only information on states but also robot's observations $z \in \mathcal{Z}$. Consequently, in this work, we define a policy summary of size n as a subset $\sigma \subset \mathcal{Z}$ of observations $\sigma = \{z_1, \cdots, z_n\}$. Each example in the summary corresponds to a demonstration of robot trajectory, $\tau_i(z_i)$, generated by executing the observation-dependent policy $\pi(\cdot)$ starting from observation z_i. These examples can be augmented with the state estimate $\hat{s}_i = f(z_i)$ and, optionally, local explanations e_i of the demonstrated behavior $\tau_i(z_i)$.

The generalized problem of policy summarization thus corresponds to selecting the subset of observations $\sigma \subset \mathcal{Z}$ of cardinality n given

- the robot's perception model $f : \mathcal{Z} \rightarrow \mathcal{S}$,
- the robot's decision-making model $g : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$,
- the training data set $\mathcal{D} \subset \mathcal{Z}$, and
- a subroutine to demonstrate $\tau(z_i)$, the corresponding state estimate $\hat{s}_i$, and (optionally) local explanation e_i,

to maximize the human's ability to accurately predict robot behavior $\pi(a|z)$ given its observation z. We next provide a solution to this human-centered problem.

IV. Out-of-Distribution Policy Summarization

As a solution to the problem of summarizing observation-dependent policies, we introduce the Out-of-Distribution Policy Summarization (OOPS) in Alg. 1. Similar to previous policy summarization methods, OOPS follows the explanation-by-example paradigm, generating key robot demonstrations to provide users with an understanding of its overall behavior π. However, instead of searching for salient examples only in its in-distribution scenarios $\mathcal{D}$, OOPS augments this dataset with out-of-distribution examples (line 7, Alg. 1). The augmented dataset $\mathcal{D} \cup \mathcal{D}_\sim$ is used to generate σ, policy summaries comprising both ID (lines 2-6) and OOD examples (lines 8-10). After selecting these observation-dependent examples $z_i \in \sigma$ of robot behavior, they are interactively presented to the user along with corresponding trajectories $\tau(z_i)$, state estimate $\hat{s}_i$, and local explanation e_i.

A. Selecting In-Distribution Examples

To accurately summarize robot behavior, ID examples are essential, even with our emphasis on OOD scenarios. These examples can aid humans in developing a ToM about the robot within limited ID settings before extending it to cover the robot's broader and potentially more complex OOD behaviors. Thus, OOPS develops summaries that include ID examples, as illustrated in lines 2-6 of Alg. 1.

1) Line 2: To select the salient ID examples, we build on existing policy summarization methods that consider state-dependent policies [11], [12], [14], [34]. In the context of

Algorithm 1 Out-of-Distribution Policy Summarization

Input: Robot's perception model $f(\cdot)$,
decision model $g(\cdot)$,
training dataset $\mathcal{D}$
size of the policy summary n
Parameter: Number of in-distribution examples, m
Output: Policy summary, $\sigma = \{z_1, \cdots, z_n\}$

1: $\sigma \leftarrow \emptyset$ ▷ *Initialize the policy summary*
2: $\sigma_{\hat{s}} \leftarrow$ IDPolicySummarization$(g, \mathcal{S}, m)$
3: **for** $i = 1 : m$ **do**
4: $\quad \hat{s}_i \leftarrow \sigma_{\hat{s}}[i]$
5: $\quad z_i \leftarrow$ SelectIDExample$(f, \mathcal{D}, \hat{s}_i)$
6: $\quad \sigma \leftarrow \sigma \cup z_i$
7: $\mathcal{D}_\sim \leftarrow$ OODDataAugmentation$(\mathcal{D})$
8: $\sigma_\sim \leftarrow$ SelectOODExample$(f, \mathcal{D}_\sim, n - m)$
9: $\sigma \leftarrow \sigma \cup \sigma_\sim$
10: **return** σ ▷ *Demonstrate the summary to the human*

Algorithm 2 SELECTIDEXAMPLE

Input: Robot's perception model $f(\cdot)$, training dataset $\mathcal{D}$, state estimate $\hat{s}_i$
Output: ID observation, z_i

$\zeta \leftarrow \{z_j \in \mathcal{D} | f(z_j) = \hat{s}_i\}$ ▷ *Select compatible ID z*
$\Delta_c \leftarrow \infty$
for $z_j \in \zeta$ **do**
  $e_j \leftarrow$ GENERATELOCALEXPLANATION(z_j, f)
  $c_j \leftarrow \Pr(\hat{s}_i | z_j, f)$ ▷ *f's confidence in $\hat{s}_i|z_j$*
  $\bar{c}_j \leftarrow \Pr(\hat{s}_i | z_j, e_j, f)$ ▷ *f's confidence in $\hat{s}_i|z_j, e_j$*
  if $c_j - \bar{c}_j \leq \Delta_c$ **then**
    $z_i \leftarrow z_j$ ▷ *Select z_j with least confidence drop*
    $\Delta_c \leftarrow c_j - \bar{c}_j$
return z_i

Algorithm 3 SELECTOODEXAMPLE

Input: Robot's perception model $f(\cdot)$, OOD examples $\mathcal{D}_\sim$, number of out-of-distribution examples k
Output: OOD observations, $\sigma_\sim$

CONFIDENCEDROPS $\leftarrow \{\}$
for $z_j \in \mathcal{D}_\sim$ **do**
  $e_j \leftarrow$ GENERATELOCALEXPLANATION(z_j, f)
  $\hat{s}_j \leftarrow f(z_j)$
  $c_j \leftarrow \Pr(\hat{s}_j | z_j, f)$ ▷ *f's confidence in $\hat{s}_j|z_j$*
  $\bar{c}_j \leftarrow \Pr(\hat{s}_j | z_j, e_j, f)$ ▷ *f's confidence in $\hat{s}_j|z_j, e_j$*
  CONFIDENCEDROPS$[z_j] \leftarrow c_j - \bar{c}_j$
$\sigma_\sim \leftarrow$ TOPK(CONFIDENCEDROPS, k, KEY=BY_HIGHEST_CONFIDENCE)
return $\sigma_\sim$

OOPS, we refer to these methods as IDPOLICYSUMMARIZATION. These methods search for salient examples within the state space rather than the observation space, assuming they are identical. *Despite this limiting assumption, we observe that they can be used to identify salient examples for summarizing the robot's decision-making model $g(a|\hat{s})$.* Hence, OOPS generates a set of m salient robot's estimated states $\sigma_{\hat{s}} = \{\hat{s}_1, \cdots, \hat{s}_m\}$ using the subroutine IDPOLICYSUMMARIZATION. This subroutine can be instantiated using any policy summarization method that considers state-dependent policies. In our experiments, OOPS utilizes AITEACHER [12] because it explicitly models human cognition, and experiments have demonstrated its ability to improve user understanding of robots across multiple tasks [17].

2) Lines 3-6: While IDPOLICYSUMMARIZATION can identify salient states, for summarizing observation-dependent policies, OOPS also needs to identify observations corresponding to those states. Formally, for each $\hat{s}_i \in \sigma_{\hat{s}}$, we seek a representative observation z_i such that $f(z_i) = \hat{s}_i$. OOPS utilizes Alg. 2 SELECTIDEXAMPLE to obtain this z_i. This subroutine searches the training dataset $\mathcal{D}$ to first find the subset of ID observations ζ compatible with $\hat{s}_i$.

To select the salient observation among this subset, we explored multiple strategies inspired by XAI algorithms for perception models [25], [26] and designed an approach based on the model's local explanations and confidence in its output. Concretely, for each observation $z_j \in \zeta$, Alg. 2 then generates a local explanation e_j of the perception model f. Existing XAI literature offers multiple algorithms for generating these local explanations [19], [20]. For our experiments, we consider image-based observations and employ saliency maps as local explanations. Saliency maps, a widely-used XAI approach, visually emphasize regions within the input image that are crucial to the perception model's predictions. Next, SELECTIDEXAMPLE computes the model's confidence in its output with and without the explanations.

The confidence without explanation c_j is computed as the logits of the perception model conditioned on the observation z_j. To compute the model's confidence given the explanation, Alg. 2 first generates a masked image $\bar{z}_j$ that retains only the regions highlighted on the saliency map e_j. Then, it computes $\bar{c}_j$ as the logits of the perception model conditioned on the masked image $\bar{z}_j$. Finally, the subroutine returns the observation $z_j \in \zeta$ with the least confidence drop: $c_j - \bar{c}_j$. *Intuitively, OOPS prioritizes the ID observation where robot's state estimates are most aligned with its local explanation to aid in user understanding of robot's ID behavior.* Fig. 2(a-b) shows selected ID observations and local explanations for the experimental task.

B. Selecting Out-of-Distribution Examples

While ID examples are important, they fail to capture robot behavior in the broader range of scenarios it will encounter in practice. Hence, incorporating OOD scenarios is equally critical. To add these OOD scenarios to the policy summary, in line 7, OOPS augments the set of observations $\mathcal{D}$ with synthetically generated observations $\mathcal{D}_\sim$. Next, in line 8, it selects the salient OOD examples from $\mathcal{D}_\sim$.

1) Line 7: Given the importance of OOD examples, a method is needed to extend the training dataset $\mathcal{D}$ with additional observations that more comprehensively cover the scenarios a robot encounters in practice. One approach is to collect additional data; in situations where this might be resource intensive, an alternative is to augment training datasets with synthetically generated data. We utilize this alternate approach in our implementation of OOPS. Data augmentation is a widely used ML technique that improves model generalization by supplementing training data with additional sources [35], [36]. Multiple strategies have been explored for data augmentation, including generative models.

While data augmentation has been employed for training robots and ML models, our novelty lies in its application to policy summarization. Building on recent advances in data augmentation [37], [38], we investigate using generative AI models to create OOD observations, denoted as $\mathcal{D}_\sim$. For our experiments, OOPS employs GPT-Image-1, a state-of-the-art image generation model to instantiate line 7. We prompt this model to produce photorealistic renderings of robot's observations, then add modifiers to enhance image diversity. These modifiers include incorporating other objects and manipulating reflections, colors, and shadows.

(a) ID: Metal Can (b) ID: Food Waste

(c) OOD: Bottle Cap (incorrectly detected by the robot as Food Waste) (d) OOD: Carton (incorrectly detected by the robot as Metal Can)

Fig. 2. Example ID and OOD observations (left), along with their saliency maps (right). OOD examples include observations where the robot incorrectly detects the task state.

Fig. 3. An instance of recycling task implemented in a university laboratory.

2) Line 8: After generating the synthetic data, OOPS selects salient OOD examples. For ID examples, we focused on selecting observations most aligned with the chosen state, thereby representing the nominal robot behavior. In contrast, for OOD examples, we prioritize observations where the robot decisions deviate the most from human expectations, thereby helping reveal edge cases of robot's behavior. Our insight is to provide a combination of ID and OOD examples that collectively summarize the robot's overall behavior. Similar problems have been explored within the explanation-by-example paradigm for image classification models [25], [26], where XAI algorithms select examples that deviate most from human expectation to improve user's model comprehension. *We extend these ideas to policy summarization.*

Specifically, through the SELECTOODEXAMPLE subroutine in Alg. 3, OOPS selects OOD observations. This subroutine selects examples from the synthetic data $\mathcal{D}_{\sim}$ rather than the training data $\mathcal{D}$. Further, to select observations, it prioritizes examples with the *highest confidence drop*. *Intuitively, these examples represent observations where the retained salient features (based on local explanations) alone are insufficient to support the original prediction, often causing the robot's decision to shift. We posit that presenting such OOD examples to human users helps reveal edge cases of robot's behavior.* Fig. 2(c-d) shows selected OOD observations and local explanations for the experimental task.

V. EXPERIMENTS

We conducted two sets of IRB-approved human subject experiments in simulated HRI tasks to evaluate the utility of OOD examples (Study #1, N=26) and the performance of the OOPS algorithm (Study #2, N=10) for policy summarization.

A. Experimental Domains

Our experimental domain considers an application in waste management, in which a robot must sort trash and recyclables. An example instance of this recycling task, implemented in a university laboratory, is depicted in Fig. 3 and demonstrated in the accompanying video. Such tasks are commonly used in robotics evaluations because it requires policies to generalize across a diverse set of objects [17], [39], [40]. Moreover, it naturally introduces unpredictable OOD objects, which can challenge the robot's policy and cause it to act unexpectedly. Our implementation of this domain features a conveyor belt (table) with six grids for object retrieval and two bins for disposal. The conveyor belt can carry up to six objects, one in each grid. Utilizing its sensors and perception models $f(z)$, the robot identifies items on the conveyor belt. Employing its actuators and decision-making model $g(a|s)$, it picks and places them into the appropriate bin.

Study #1 instantiates this task with abstract objects (red and green blocks) and a rule-based perception model, thereby evaluating policy summarization in HRI scenarios where participants have no prior knowledge about the domain (e.g., which objects are recyclable). In contrast, simulating situations where users may possess prior knowledge, Study #2 instantiates the recycling task with four object types: metal cans, paper cups, food waste, and styrofoam. Metal cans and paper cups are categorized as recyclable, while food waste and styrofoam are non-recyclable. Both implementations of the domain include 78125 possible ID states $\hat{s}$, with *unlimited* OOD observations z. The action space is of size 9: the robot may pick up an object from any of the six grid spaces, drop an object into either one of the two bins, or stay still.

The robot perceives the conveyor belt using its camera, which provides image-based observations z. Given an image, the perception model $f(z)$ estimates the task state $\hat{s}$, which encodes the object in each grid. Study #1 utilizes a rule-based perception model, while Study #2 utilizes a fine-tuned YOLOv12n network [41]. Mirroring real-world deployment, these perception models can make errors and incorrectly detect objects. The decision-making model, $g(a|s)$, is trained using value iteration [33]. The robot prioritizes picking metal cans, then paper cups, food waste, and styrofoam. If multiple objects of the same type are detected, it picks the one with the smallest grid number first. The robot places items detected as recyclable into Bin 1, and others into Bin 2. If no objects are detected, the robot remains stationary.

B. Mechanisms for Delivering the Policy Summaries

Just as robot policies are often trained in simulation due to resource constraints, human users are frequently trained to use robots in simulated environments [17]. Further, HRI researchers commonly validate robot explanation techniques

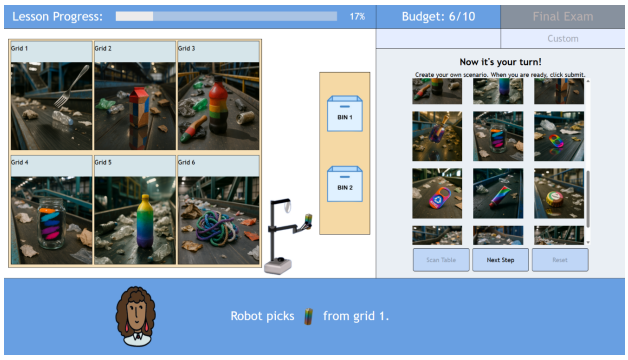


Fig. 4. Interactive user interface used to deliver policy summaries to human participants. Please see the accompanying video for a demonstration.

using on-screen visualizations or simulated robots [11]–[13]. Similarly, policy summaries are delivered to the participants via a user interface (UI, illustrated in Fig. 4) in our experiments. Through the UI, participants can request examples of the robot's ID and OOD behaviors. In both studies, examples of ID behavior are generated using the IDPolicySummarization subroutine of OOPS and are referred to as AI lessons in the UI. Following [17], after each AI lesson, participants received a *quiz question* about the robot's behavior in ID scenarios designed to reinforce their learning.

The approach to generating OOD examples differs across the two studies. Study #1 is designed to evaluate the utility of OOD examples, irrespective of their source, and uses a hand-crafted set of OOD examples. Study #2 is designed to evaluate the feasibility of algorithmically generating effective OOD examples and uses the OOPS algorithm to generate OOD examples. In both studies, the participant is given a set of OOD images, which she can use to generate custom scenarios (e.g., by placing the objects on conveyor belts) and request an example of robot behavior in these scenarios. These examples of OOD behavior are referred to as custom lessons in the UI.

For the custom lessons, the UI includes two interactive buttons: one ("Scan Table") to view how the robot perceives the objects $\hat{s} = f(z)$ and another ("Next Step") to view the robot's behavior $a \sim g(a|s) = \pi(a|z)$. Both ID and OOD examples are augmented with local explanations e, as illustrated in Fig. 2. In our implementation, we utilize saliency maps as local explanations [20]. The saliency maps (right) highlight the visual features in the corresponding images (left) that the robot's perception model considers important. The UI (bottom) also includes a textual description of robot behavior in the selected ID and OOD example.

C. Experiment Design, Procedure, and Metrics

In both studies, participants first provided informed consent and completed a short demographic survey. They were then guided through a simplified two-object training task, which included a tutorial on the experiment and its UI. Following this, participants were asked to use the UI to learn about the robot's perception and decision-making behavior in the complex experimental task. In both studies, participants had a budget of ten examples, i.e., $|\sigma| = n = 10$.

Study #1 adopted a between-subjects design. Both groups received identical five AI lessons, i.e., $m = 5$. However, the control group was only offered ID objects in custom lessons, while the experimental group was given additional OOD objects. We conducted this study as a randomized controlled trial, recruiting 26 participants from Rice University. The average age of participants was 21.07, with 17 females and 9 males. Of these participants, 25 reported having previously used AI and robots, with 3 of them having programmed AI and robots.

Study #2 was conducted as an iterative prototyping study to both evaluate OOPS and inform the design of future policy summarization methods. All participants received identical ID and OOD objects generated using OOPS. We conducted this experiment by recruiting 10 participants (6 male, 4 female) from Rice University, with their ages ranging from 20 to 29. Four participants had experience developing robots, and 8 participants reported using AI regularly.

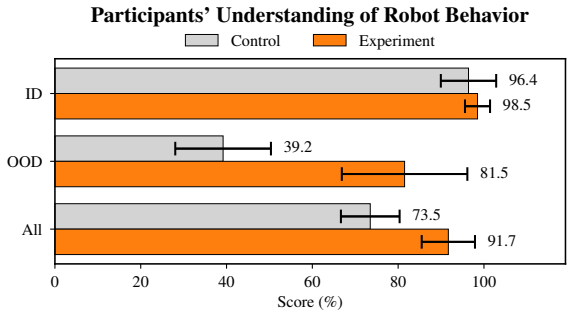
Participants' understanding of the robot's behavior, including perception and decision-making for both ID and OOD objects (both seen and unseen during training), was evaluated through a multiple-choice final exam. Questions took two forms: 1) predicting $\hat{s} \rightarrow a$ and $z \rightarrow a$ with answers $a \in \mathcal{A}$ and 2) $z \rightarrow \hat{s}$, where participants identify objects that the robot prioritizes. Results were calculated as percentage of questions answered correctly. Table I lists the types and number of questions used in Study #2. After the exam, participants completed a post-experiment survey regarding their learning experience, learning preferences, and the influence of OOD objects on their learning.

D. Experimental Results

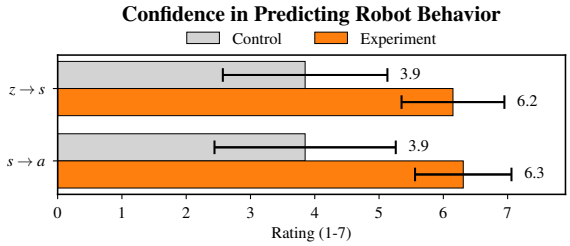
We now report the results of both studies, which together highlight the importance of OOD examples and the promising performance of the OOPS algorithm in generating them.

1) Policy summaries with OOD examples improve humans' ability to predict robot behavior, especially in OOD scenarios: Through a between-subject design in Study #1, we examine the impact of deploying OOD objects for policy summarization. A Mann–Whitney U test was conducted to compare the performance of the experimental group (who received policy summaries containing both ID and OOD objects) with the control group (who had access only to ID objects). Note that while the experimental group is exposed to OOD objects during training, the final exam includes unseen OOD objects which do not appear in the training ensuring fair comparison for both groups. We present the scores (%) for both groups in Fig. 5a.

Participants in the experiment condition scored statistically significantly higher on the final exam (M = 91.70%, SD = 1.55) compared to the control group (M = 73.53%, SD = 1.71), with $U = 2$, $p < 0.001$. The notable difference in participants' scores for OOD questions, where the experimental group scores almost twice as much as the control group, suggests that humans who are given policy summaries with OOD objects are better able to predict the robot behavior π than those who are only given ID policy summaries.



(a) Objective results from the final exam.



(b) Participants' self-reported confidence in predicting robot perception $f(z)$ and decision-making $g(a|s)$ on a 7-point scale.

Fig. 5. Objective and self-reported results from Study #1.

Through the post-experiment survey, participants also rated their confidence in predicting the robot behavior. As shown in Fig. 5b, participants in the experimental group reported higher confidence in predicting both robot's perception and decision-making behavior.

2) OOPS is capable of generating useful summaries, composed of both ID and OOD examples, for observation-dependent policies: Having confirmed the utility of OOD examples, we proceed to Study #2 to evaluate the OOPS algorithm's ability to generate effective OOD examples and policy summaries. As in the first study, participants' final exam scores are used to measure their understanding of the robot's perception and decision-making. However, this study differs from Study #1 by employing photorealistic objects, a more complex perception model, and OOPS to synthetically generate OOD examples. Table II presents participants' scores (in %) across different question types.

Participants demonstrated a strong understanding of the perception model $f : z \rightarrow \hat{s}$, achieving the high score of 85.9% after reviewing OOPS-generated policy summaries. When predicting robot actions (a), users showed higher accuracy on seen states during lessons compared to unseen samples. Scores were 85.0% vs. 67.5% for ID objects, and 67.5% vs. 40.0% for OOD objects. Furthermore, predicting robot behavior on OOD objects (considering both $f : z \rightarrow \hat{s}$ and $\pi : z \rightarrow a$) proved more challenging than on ID objects, with scores of 70.3% and 80.7%, respectively.

Through the post-experiment survey, participants also subjectively rated the examples generated by the OOPS algorithm. Overall, participants rated the experience of learning about robot behavior using OOPS as 5.1 on a scale of 1-7, with 5 representing "Good." Collectively, these findings indicate

TABLE I

TYPE AND NUMBER OF QUESTIONS IN THE STUDY #2 FINAL EXAM. SEEN/UNSEEN DENOTE z, $\hat{s}$, AND a THAT PARTICIPANTS HAVE/HAVE NOT SEEN VIA σ.

Question Type		$z \rightarrow \hat{s}$	$\hat{s} \rightarrow a$	$z \rightarrow a$	*Total*	
ID	Seen	3	4	–	7	14
	Unseen	3	4	–	7	
OOD	Seen	3	–	4	7	14
	Unseen	3	–	4	7	
		12	8	8		28

TABLE II

OBJECTIVE RESULTS FROM THE STUDY #2 FINAL EXAM (%). SEEN/UNSEEN DENOTE z, $\hat{s}$, AND a THAT PARTICIPANTS HAVE/HAVE NOT SEEN VIA σ.

Question Type		$z \rightarrow \hat{s}$	$\hat{s} \rightarrow a$	$z \rightarrow a$	*Average*	
ID	Seen	80.0	85.0	–	82.5	80.7
	Unseen	90.0	67.5	–	78.8	
OOD	Seen	86.7	–	67.5	77.1	70.3
	Unseen	86.7	–	40.0	63.4	
		85.9	76.3	53.8		75.5

that OOPS effectively aids users in comprehending the robot's visual perception model $f(\cdot)$ through the generation and selection of informative OOD examples. Nevertheless, anticipating the robot's decision-making behavior in novel samples, especially those with OOD objects, remains a hurdle for participants and an area requiring additional research in policy summarization.

3) Participants find both ID and OOD examples important for understanding robot behavior: In a post-experiment survey, participants rated their agreement with the statement "ID/OOD items were important for me in learning about the robot's behavior." on a 7-point scale (1 = Strongly Disagree, 7 = Strongly Agree). Across both studies, participants consistently rated both ID and OOD examples as important. In the first study, ID objects were rated 6.26 ± 1.1 and OOD objects 6.50 ± 0.8. In Study #2, ID objects were rated 5.70 ± 0.9 and OOD objects 5.20 ± 1.6. Participants' open-ended responses further confirm the necessity of including both ID and OOD examples in policy summaries, as one noted:

> "To accurately understand the robot's behavior, you need to know how the robot will respond when presented with a variety of items, not just the ones used during training (the in-distribution items). Because we are not given the actual implementation of the robot's behavior, understanding how the robot behaves for in-distribution items and out-of-distribution items is important."

Another participant further highlighted the relevance of OOD objects for a key downstream use case of OOPS, debugging and detecting unexpected robot behavior, stating:

> "In-distribution is good for understanding the patterns, but out of distribution is good for figuring out what happens when there is a problem or an outlier. This is good for predicting unexpected behavior."

Thus, the objective and subjective results of the experiments collectively confirm that both ID and OOD objects are essential and should be thoughtfully selected in policy summaries to support users' comprehension of robot behavior.

VI. Conclusion

This paper extends the problem of robot policy summarization to consider observation-dependent policies. We propose OOPS: Out-of-Distribution Policy Summarization that generates salient examples of robot behavior in both in- and out-of-distribution scenarios. Two studies demonstrate that incorporating OOD examples improves human participants' understanding of robot behavior and that OOPS produces useful policy summaries. Our experiments show that humans engage with OOD examples, and both ID and OOD examples are essential for effective policy summarization.

Limitations and Future Work: Our work is not without limitations. While we consider more complex policies that utilize visual input than prior art, OOPS is limited to modular policies. Extending it to end-to-end policies is an immediate direction of future work. Further, evaluations are limited to a single domain. While this domain is complex, we welcome reproducibility studies evaluating OOPS in other domains. Lastly, our work highlights the potential of data augmentation to enhance explainability of robot behavior.

References

[1] A. Thomaz, "Robots in real life: Putting HRI to work," in *IEEE/ACM Int. Conf. on Human-Robot Interaction*, pp. 3–3, 2023.

[2] R. Murphy, "Rescue robots," in *Robotics Goes MOOC*, Springer, 2025.

[3] L. Orlov-Savko, Z. Qian, G. Gremillion, C. Neubauer, J. Canady, and V. Unhelkar, "RW4T Dataset: Data of human-robot behavior and cognitive states in simulated disaster response tasks," in *IEEE/ACM Int. Conf. on Human-Robot Interaction*, 2024.

[4] P. Qian, F. Bajraktari, C. Quintero-Pena, Q. Meng, S. Hamlin, L. Kavraki, and V. Unhelkar, "ASTRID: A Robotic Tutor for Nurse Training to Reduce Healthcare-Associated Infections," in *Robotics: Science and Syst.*, 2025.

[5] A. Thomaz, G. Hoffman, and M. Cakmak, "Computational human-robot interaction," *Foundations and Trends® in Robotics*, vol. 4, no. 2-3, pp. 105–223, 2016.

[6] S.-l. Lee, I. Y.-m. Lau, S. Kiesler, and C.-Y. Chiu, "Human mental models of humanoid robots," in *IEEE Int. Conf. Robot. Autom.*, pp. 2767–2772, 2005.

[7] T. Hellström and S. Bensch, "Understandable robots-what, why, and how," *Paladyn, J. of Behavioral Robotics*, pp. 110–123, 2018.

[8] M. M. De Graaf, B. F. Malle, A. Dragan, and T. Ziemke, "Explainable robotic systems," in *IEEE/ACM Int. Conf. on Human-Robot Interaction*, pp. 387–388, 2018.

[9] S. Anjomshoae, A. Najjar, D. Calvaresi, and K. Främling, "Explainable agents and robots: Results from a systematic literature review," in *Int. Conf. on Autonomous Agents and Multiagent Systems*, 2019.

[10] O. Amir, F. Doshi-Velez, and D. Sarne, "Summarizing agent strategies," *Int. Conf. on Autonomous Agents and Multiagent Systems*, vol. 33, no. 5, pp. 628–644, 2019.

[11] S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan, "Enabling robots to communicate their objectives," *Autonomous Robots*, vol. 43, 2019.

[12] P. Qian and V. Unhelkar, "Evaluating the role of interactivity on improving transparency in autonomous agents," in *Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 1083–1091, 2022.

[13] M. Lee, R. Simmons, and H. Admoni, "Improving the transparency of robot policies using demonstrations and reward communication," *ACM Trans. on Human-Robot Interaction*, 2025.

[14] D. Amir and O. Amir, "Highlights: Summarizing agent behavior to people," in *Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 1168–1176, 2018.

[15] O. Watkins, S. Huang, J. Frost, K. Bhatia, E. Weiner, P. Abbeel, T. Darrell, B. Plummer, K. Saenko, and A. Dragan, "Explaining robot policies," *Applied AI Letters*, vol. 2, no. 4, 2021.

[16] P. Qian, H. Huang, and V. Unhelkar, "PPS: Personalized policy summarization for explaining sequential behavior of autonomous agents," in *AAAI/ACM Conf. on AI, Ethics, and Society*, pp. 1167–1179, 2024.

[17] P. Qian and V. V. Unhelkar, "Interactively explaining robot policies to humans in integrated virtual and physical training environments," in *IEEE/ACM Int. Conf. on Human-Robot Interaction*, 2024.

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *NeurIPS*, vol. 30, 2017.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *IEEE ICCV*, pp. 618–626, 2017.

[21] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *Int. Conf. on Machine Learning*, pp. 5338–5348, 2020.

[22] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *NeurIPS*, vol. 33, pp. 20554–20565, 2020.

[23] F. I. Doğan, G. I. Melsión, and I. Leite, "Leveraging explainability for understanding object descriptions in ambiguous 3D environments," *Frontiers in Robotics and AI*, vol. 9, p. 937772, 2023.

[24] Y. Kim, H. Chen, S. Alghowinem, C. Breazeal, and H. W. Park, "Joint engagement classification using video augmentation techniques for multi-person human-robot interaction," in *Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 698–707, 2023.

[25] Y. Rong, P. Qian, V. Unhelkar, and E. Kasneci, "I-CEE: tailoring explanations of image classification models to user expertise," in *AAAI Conf. on Artificial Intelligence*, vol. 38, pp. 21545–21553, 2024.

[26] S. C.-H. Yang, W. K. Vong, R. B. Sojitra, T. Folke, and P. Shafto, "Mitigating belief projection in explainable artificial intelligence via Bayesian teaching," *Scientific Reports*, vol. 11, no. 1, p. 9863, 2021.

[27] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *IEEE Symp. Series on Computational Intelligence (SSCI)*, 2020.

[28] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[29] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable Markov decision processes in robotics: A survey," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 21–40, 2022.

[30] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. of Robotics Research*, p. 02783649241273668, 2023.

[31] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, *et al.*, "OpenVLA: An open-source vision-language-action model," in *Conf. on Robot Learning*, pp. 2679–2713, 2025.

[32] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[33] R. S. Sutton, A. G. Barto, *et al.*, *Reinforcement learning: An introduction*. MIT Press Cambridge, 2018.

[34] M. S. Lee, H. Admoni, and R. Simmons, "Machine teaching for human inverse reinforcement learning," *Frontiers in Robotics and AI*, vol. 8, p. 693050, 2021.

[35] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. of Big Data*, pp. 1–48, 2019.

[36] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann, "Data augmentation can improve robustness," *NeurIPS*, pp. 29935–29948, 2021.

[37] C. Shivashankar and S. Miller, "Semantic data augmentation with generative models," in *CVPR*, pp. 863–873, 2023.

[38] C. Zheng, G. Wu, and C. Li, "Toward understanding generative data augmentation," *NeurIPS*, vol. 36, pp. 54046–54060, 2023.

[39] D. Ghose, M. A. Lewkowicz, K. Gezahegn, J. Lee, T. Adamson, M. Vazquez, and B. Scassellati, "Tailoring visual object representations to human requirements: A case study with a recycling robot," in *Conf. on Robot Learning*, vol. 205 of *PMLR*, pp. 583–593, 2023.

[40] A. Herzog, K. Rao, K. Hausman, Y. Lu, P. Wohlhart, *et al.*, "Deep RL at scale: Sorting waste in office buildings with a fleet of mobile manipulators," in *Robotics: Science and Syst.*, 2023.

[41] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *NeurIPS*, vol. 38, pp. 78433–78457, 2026.