



Endowing Robots with Longer-term Autonomy by Recovering from External Disturbances in Manipulation Through Grounded Anomaly Classification and Recovery Policies

Shuangqi Luo¹ · Hongmin Wu² · Shuangda Duan³ · Yijiong Lin⁴ · Juan Rojas⁵

Received: 24 August 2020 / Accepted: 4 January 2021 / Published online: 22 February 2021
© The Author(s) 2021

Abstract

Robots are poised to interact with humans in unstructured environments. Despite increasingly robust control algorithms, failure modes arise whenever the underlying dynamics are poorly modeled, especially in unstructured environments. We contribute a set of recovery policies to deal with anomalies produced by external disturbances. The recoveries work when various different types of anomalies are triggered any number of times at any point in the task, including during already running recoveries. Our recovery critic stands atop of a tightly-integrated, graph-based online motion-generation and introspection system. Policies, skills, and introspection models are learned incrementally and contextually over time. Recoveries are studied via a collaborative kitting task where a wide range of anomalous conditions are experienced in the system. We also contribute an extensive analysis of the performance of the tightly integrated anomaly identification, classification, and recovery system under extreme anomalous conditions. We show how the integration of such a system achieves performances greater than the sum of its parts.

Keywords Failure detection and recovery · Cooperative manipulators · Human-centered robotics · Learning from demonstration

✉ Juan Rojas
juan.rojas@cuhk.edu.hk

Shuangqi Luo
sk.law.lsq@gmail.com

Hongmin Wu
hongminwu0120@gmail.com

Shuangda Duan
470963660@qq.com

Yijiong Lin
yijionglin.bourne@gmail.com

- ¹ University of Maryland, College Park, MD, USA
- ² Guangdong Institute of Intelligent Manufacturing, Guangzhou, People's Republic China
- ³ University of Waterloo (Foshan) Innovation Center, Foshan, People's Republic China
- ⁴ University of Bristol, Bristol, UK
- ⁵ School of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong, People's Republic China

1 Introduction

1.1 Problems

As robots experience increased levels of unstructured environments and shared workspaces with humans, so will the possibility of anomalies and failure. Even as manipulation and control algorithms become increasingly robust, failure modes continue to exist.

Numerous sources of failure and execution anomaly arise from the complex dynamics found in robots, limited modeling ability, and robot's interactions with the world. We specify anomalies to be executions whose sensor signatures deviate from a robot's learned expected models.¹ Sources of anomaly include: (i) internal errors that can result from improper modeling of visual, kinematic, or dynamic models

¹This does not imply that a robot's learned expected models always align with success. A robot's observation may confirm a learned model and yet fail in actuality. This would equate to a misjudgment on the part of a human. Evaluating a model's reciprocity with true events in the world is an important open problem but outside the scope of this work

and (ii) external anomalies that arise from the inability to model accidental collisions (human-robot, robot-world, or robot-object-world), object slips due to inertial dynamics, misgrasps; or even a chain reaction where one anomaly triggers other anomalies. Figure 1 illustrates two examples of possible anomalies in a kitting experiment. Anomalous conditions are also hard to model as similar anomalies can occur with wide variability, making it challenging for robots to recognize. Recovery system performance will depend significantly on the ability to properly identify and understand the nature of the anomaly.

Under the recovery domain, there are many other considerations: am I recovering from an accidental error or a persistent one? Can I leverage different recovery policies without deleterious effects between them? Can the recovery policies work consistently when the robot experiences repeated and strenuous anomaly conditions? Are any parts of the task more susceptible to failure? Can my recovery system overcome anomalies triggered whilst a recovery skill is executing?

1.2 Related Works

Few papers have studied the development of explicit recovery policies for recovery of anomalous conditions, especially those that are characterized by random or unstructured qualities that are hard to model or anticipate.

1.2.1 Manipulation Skills

Recovery action design shares fundamentals with skill learning and skill sequencing, where there has been much ongoing research in the last two decades. Ijspeert et al. introduced goal-directed dynamic motion primitives (DMPs) that could learn arbitrarily complex shaped skills from one demonstration [1]. Paraschos et al. conceived probabilistic motion primitives where demonstrations were approximated by a linear combination of weighted basis functions, where the weight vector was modeled by a distribution [2]. Other probabilistic methods derive the regression function instead of learning it [3]. Beyond skill bootstrapping, incremental corrections of motions has been another branch of study that has been studied and one that closely connects to this work. In [4], Jain et al. use human demonstrators to iteratively provide trajectories that are used to update the encoded skill via co-activations. In [5], Konidaris et al. learn to build skill trees by segmenting a demonstration into skills with specific goals and which are encoded with suitable abstractions. These abstractions are improved via policy learning and finally merged into skill trees. In [6], Gutierrez et al. design a method that takes a given demonstration and selects between possible model updates by either adding new nodes or new transitions in the FSM. In [7], focuses on iteratively



Fig. 1 Anomaly examples when a robot performs a kitting experiment in a shared workspace with a human. Top: an accidental collision between the robot and the human as the human places a new object in the collection area. Bottom: a tool collision as the end-effector prepares to pick an object

improving skill learning by only focusing on one-feature-at-a-time rather than all features. None of these works consider how corrections could take place in view of the presence of (a myriad of) anomalies that could drastically alter the state of the skill and the environment in which the robot is working. To begin to think about correcting robot skills in this context, one must first have the ability to understand anomalies.

1.2.2 Anomaly Identification and Classification

We define anomalies as the deviation of sensor-related signatures from those experienced in nominal executions. In this work, we abuse terminology some and use anomalies and failure interchangeably. Evidently, not all anomalies need lead to failures). Anomalies in robotic environments have been studied particularly in structured and uni-modal formats [8, 9]. More recently, anomaly identification and classification in unstructured environments have been studied. The majority of these are based on Hidden Markov Models (HMMs) and differ in the way they try to learn the HMMs parameters. In [10], language is used to generate motion, and a simple HMM is used to detect success or failure based on trajectory position information. In [11], visual, audio, and proprioceptive features are used through Hidden Markov Models (HMMs) and other heuristics on a tabletop task to detect failure. In [12], a hierarchical Dirichlet process (HDP) prior was used with HMMs and a Gaussian observation model and Gibbs sampling to do anomaly iden-

tification and multi-class classification. This work used simple threshold based to do flag anomalies. In [13], knowledge-based process models are combined with HMMs. Operation-invariant characteristics such as screwdriving mechanics and stage transition graph, support generalization across different screwdriver sizes that have varying tightening current, motor velocity, insertion force, and tightening force. The knowledge-based models, diminished the need for data and labeling. In [14], the sticky version of HDP was used along a vector autoregressive observation model to do anomaly identification and multi-class classification with a dynamic threshold that updates its parameters according to the latent state of a sub-task. In [15], anomaly identification was conducted with HMMs whose detection threshold varied according to clusters of execution progress. In [16], the work was improved by seeking to eliminate discontinuities between clusters and opted to use a Gaussian-process regressor function. In [17], built on top of the sHDP-HMM models to design a more robust anomaly identification method that leverages the gradient of the forward model's log-likelihood achieving robustness even in recovery scenarios. In [18], an artificial neural network was used to identify and classify anomalies. The network used the latent HMM states as input features along with a convolutional neural network for image input.

1.2.3 Recovery

Few works design recovery policies that explicitly handle the occurrence of various anomalies at different times in a task. For example, in [19, 20], the entire task is only re-attempted upon failure. In [21], Chang et al. devised an error recovery system based on Petri Nets learned from demonstration. Error conditions however were defined based on object location: if objects were not located in expected states, an error was triggered. This forced the system to maintain a growing list of expected object locations. The work did not consider other anomaly sources. In [22], failure classification was performed for only one perturbation and it was pre-taught. No failure identification was presented nor was there an explicit recovery policy. Instead, a recovery behavior was inserted manually in a specific place in the task and no explicit experimental results quantified recovery versatility and robustness. Likewise the ability to grow recovery behaviors incrementally over time was absent. In [23], a system that allows for the incremental addition of skills is taught, but there is no mentions on how anomalies could be explicitly classified. Adaptive behavior was taught for two anomalies that occurred predictably and were characterized by a consistent structure. No explicit recovery policy was presented in this work to handle anomalies. In [24], proposes an approach to improve the robustness

of a policy learned from demonstration for contact tasks. Contact is modeled via a multimodal state transition function. Then, a reinforcement learning-based policy is learned to optimize skill and recovery skill selection. The policy is incrementally improved by expanding the action space by generating recovery skills with a heuristic. In this work, the environment is assume to be quasi-static, failures are not explicitly classified—they primarily arise in the form of obstructions, so the policy learns to re-route around them. The work is demonstrated via three simulation problems. While the work is promising, it is questionable whether it can scale to unstructured scenarios, especially where there is collaboration with humans, and a diverse set of errors occur.

As the literature stands, the works that have carried out recovery with some anomaly identification or classification used a fixed point in the task to return to upon failure sometimes resulting in very inefficient policies [20]. The works that address recovery more directly do not explicitly consider anomaly classification; as such, they address single error-types under similar conditions. There is a lack of work studying how recoveries look like with various types of anomalies at different points in the task. No study considers how anomalies with different “eccentricities” (temporary vs. persistent natures) could use different types of recovery policies (simpler and more complex) to resolve their consequences more efficiently and how different policies influence each other when enacted in an interleaved fashion throughout the task. Existing research also does not provide an in-depth study of whether the policies work equally well at different points in time of the task? That is, are some parts of the task more susceptible or fragile than other parts? Furthermore, what if anomalies re-occur repeatedly? How much strain can the recovery framework resolve? An interesting case here is how the system responds to cascaded-anomalies? That is anomalies, that trigger anomalies. Can the system recognize them and recover from them? Another consideration that has been ignored is how do these recovery (and anomaly identification and classification) systems work during an ongoing recovery? That is, can we continue to effectively identify, classify, and recover when anomalies take place in the midst of a recovery effort? This is an important consideration because it is imperative that the system has an ongoing introspective ability at all times. Finally, the works that do consider anomaly identification and classification lack a comprehensive study of the interdependence of these systems and their combined effect.

1.3 Contributions

The research question that we studied in this work is: “Given the ability to classify anomalies, to what extent can we

extend long-term autonomy when using a simple set of task-level recovery policies that grow over time”.

In this paper we make policy contributions, analytical contributions, and resource contributions as listed below:

- 1) The learning of contextual recovery policies that provide unique recoveries for both accidental and persistent anomalies at specific locations in the task graph.
- 2) The ability to recover (and introspect) reliably whilst the robot executes a recovery behavior triggered by a previous anomalous condition.

We believe that we also offer contributions in the knowledge created by two extensive analysis as follows:

- 1) An extended performance analysis of the anomaly identifier and multi-class anomaly classification (aided by improved variational inference) across more than 700 trials that take place everywhere during the nominal task, but also extensively during recovery skills in real-robot experiments.
- 2) The knowledge that arises by contrasting the joint and disjoint performance of the anomaly identifier, classifier, and recovery systems through comprehensive experimentation and evaluation across more than 700 trials showing that when studied together the system is able to self repair when incorrect classification and recovery policies are originally selected.

We also claim a novel dataset and open-source code framework with details as follows:

- 1) An extensive anomaly dataset of a co-bot Kitting experiment that consists of 538 rosbags with multimodal-sensory, RGB, and motor information across a wide range of anomalous conditions and recoveries that take place all throughout the task different parts of the task (details in Supplement 2). The code is completely open-sourced, including the entire manipulation-introspection-and recovery framework built on ROS and python.

1.4 Relation to Previous Work

Our contribution builds on top of our previous introspection system which could introspect into nominal skills and identify anomalies, but not classify them or recover contextually from them [25]. We also use our improved anomaly identification method presented in [17]. In this paper, we significantly expanded introspection to deal with the online classification of a challenging set of anomalies with faster and more accurate inference memoized variational infer-

ence with scalable adaptation as the modeling mechanism. In this paper we also offer better recovery policies than the naïve recovery policy design of [20]. The policy presented can only address one-off accidental anomalies, and often does so inefficiently. In this work, we improve how we resolve accidental anomalies and also introduce a new contextual recovery policy for persistent anomalies that cannot be resolved with the previous policy.

1.5 Methodology

Once an anomaly is classified, a recovery policy is executed. Recovery policies include re-enacting or adaptive policies. Re-enacting policies resolve accidental (one-off) anomalies, while adaptive policies resolve persistent anomalies. Re-enactments re-attempt either a current or previous manipulation skill but with new goal parameterizations. Re-enactments are learned from human users by modeling human recovery choices through a multinomial distribution of task nodes. Once learned, new node transitions are introduced in the graph for specific accidental anomalies at specific nodes. For adaptive policies, the robot requires user intervention to provide skill training to overcome a persistent anomaly at a given point in the task graph. Once an adaptive recovery is trained (including both skill generation and introspection models), it is introduced into the graph while retaining previously learned policies from the parent node. The approach fashions a system that incrementally learns anomalies globally and recoveries contextually (Section 5).

1.6 Experiments and Results

A co-bot experiment performing kitting tasks is used as a proof-of-concept. Anomaly identification across more than 700 trials were recognized with 93.09% accuracy; multi-class classifications with 96.15% accuracy, and as for recoveries, we recovered consistently and reliably. In particular, 82% across all our anomaly scenarios and 95% in more typical scenarios. Of significance was the system’s ability to self-correct. In situations where a misclassification was triggered and consequently an incorrect recovery policy, the system quickly self-corrected and completed the task successfully. The current framework has broad applicability to all manipulation domains that suffer from uncertainties in unstructured environments: making industrial and service robots prime candidates for this technology. Supplemental information found at [26] include video, dataset, results and analysis, and robot-agnostic source-code for the co-bot kitting experiment with anomalies and recovery information.

2 Overview

In this section we introduce a system overview. A summary of all notation is found in Appendix C, which is accessible through our supplementary page [26]

Directed graphs are a useful tool to manage complexity in manipulation tasks [22, 23, 27]. Motion comprises structure, not unlike that of grammar, encoded as a set of motion primitives and associated sensory-motor perceptions [28–30]. Consider a graph \mathcal{G} that consists of a sequence of behavior nodes that form a task. Behaviors \mathcal{B} , in turn, are composed as either simple or compounded actions; where actions are represented by nodes \mathcal{N} . Actions are connected by transitions \mathcal{T} ; as such, behaviors too are connected by transitions. A node transition from a node \mathcal{N}_s to another node \mathcal{N}_t is denoted as: $\mathcal{T}_{s,t} = \{s, t \in \mathcal{N}\}$ (we can extend this notation to longer chains). The manipulation graph is thus the set of nodes and transitions $\mathcal{G} : \{\mathcal{N}, \mathcal{T}\}$. We also wish to cement the idea that original behaviors that the task was bootstrapped with are referred to as *milestones* $\mathcal{B} = (\mathcal{B}_1, \dots, \mathcal{B}_i)$ and denote a fixed fundamental set of behaviors.

Recovery policies, more concretely adaptive policies, will generate new actions \mathcal{N}_{ij} as transitions from a current node to the new node (see Section 5.2). The node insertion creates a branch. Such new nodes will always have their successor be the next milestone (e.g. see the *rec_mvz_anom_k* node in Fig. 2). Furthermore, our system allows recoveries-over-recoveries when anomalies \mathcal{F} occur during an active recovery behavior. As such, new adaptive nodes \mathcal{N}_{ijk} are generated (see the *rec_rec_mvz_anom_m* node in Fig. 2). In this way, a set of nodes in a task, those within milestone behaviors and those in branches $\mathcal{N} = \{\mathcal{N}_i \cup \mathcal{N}_{ij} \cup \dots \cup \mathcal{N}_{ij\dots q}\}$ can incrementally grow over time as new capabilities are introduced.

Nodes do more than just generate motion. They consist of parallel modules that encapsulate a wide range of functions (skill generation, introspection, visual goal setting, language processing, navigation, etc). For this work, we restrict node modules to: skill generation \mathcal{S} , visualization \mathcal{V} , and introspection \mathcal{M} . In a given task, skill modules $\mathcal{S}_m = \{S_1, \dots, S_M\}$ perform the necessary motor skills to achieve a task (Section D). Visualization modules $\mathcal{V}_m = \{V_1, \dots, V_M\}$ process goals for specific motor skills (Section 3). Introspection models² $\mathcal{M}_m = \{M_1, \dots, M_M\}$, similar to process monitoring in the literature, aid a robot to understand the types of skills or anomalies that are experienced within a

task (Section 4). We generate and maintain skill, visual, and anomaly libraries on a per-task basis.³

The introspection module triggers anomaly flags when the system experiences sensory-motor signatures that deviate from the nominal signals expected in the currently running node (anomalies may not necessary entail true failures). Once an anomaly is triggered, an anomaly label \mathcal{F}_x is produced. Classifying anomalies is harder than classifying nominal skills because the variability under which anomalies occur is much larger (see Section 4). Similarly, acquiring anomalous data is more challenging. Discovering anomalies in a task is not a straight forward process and deciding on how to discriminate between them is hard. The policy under which anomalies are re-generated can be controversial: should they be induced or only expected to occur accidentally (Section 2.2 comments further on these issues).

After classification recovery actions \mathcal{R} (re-enactment policies \mathcal{R}_R or adaptive policies \mathcal{R}_A) are used. Re-enactments resolve one-off (accidental) anomalies. Adaptations resolve persistent anomalies. Re-enactments re-attempt a selected same or previous optimal skill. The policy transitions from the current node \mathcal{N}_i to the designated goal node \mathcal{N}_g such that $\mathcal{R}_R : \mathcal{T}_{\mathcal{N}_i, \mathcal{N}_g}$ (see Section 5.1). Adaptations, on the other hand, request user intervention to overcome the persistent fault \mathcal{F}_x and are inserted into the graph such that: $\mathcal{R}_A : S_{\mathcal{N}_i} | \mathcal{F}_x \rightarrow N_{ij}(Sm, Mm, Vm)$. Adaptations are incrementally introduced to the system as persistent anomalies appear (see Section 5.2).

2.1 Experimental Setup

A co-bot-based Kitting experiment was selected to test the recovery policies. We present the experimental testbed and manipulated objects, as well as, the details regarding external disturbances and data collection procedure.

2.1.1 Kitting Experiment

The collaborative kitting experiment consists of a robot and a human co-worker that closely collaborate to place a set of goods in a packaging box. The human co-worker places objects in the robot’s “collection bin” (located at the front of the robot) in a one-at-a-time fashion (see Fig. 4a). The objects may accumulate in a queue in front of the robot. As soon as the first object is on the table, the robot identifies the object and begins the placing process in the packaging box located to the right of the robot. Thus, the robot picks an

²We abuse the term and refer to introspection as the physical understanding of what the robot is doing and how. As such, the aim is beyond pure process monitoring; though in this work, we do not discuss the “how” component.

³We do not explicitly study the re-use of skills or introspection models across tasks—this too is critical in the re-usability and scalability of these systems

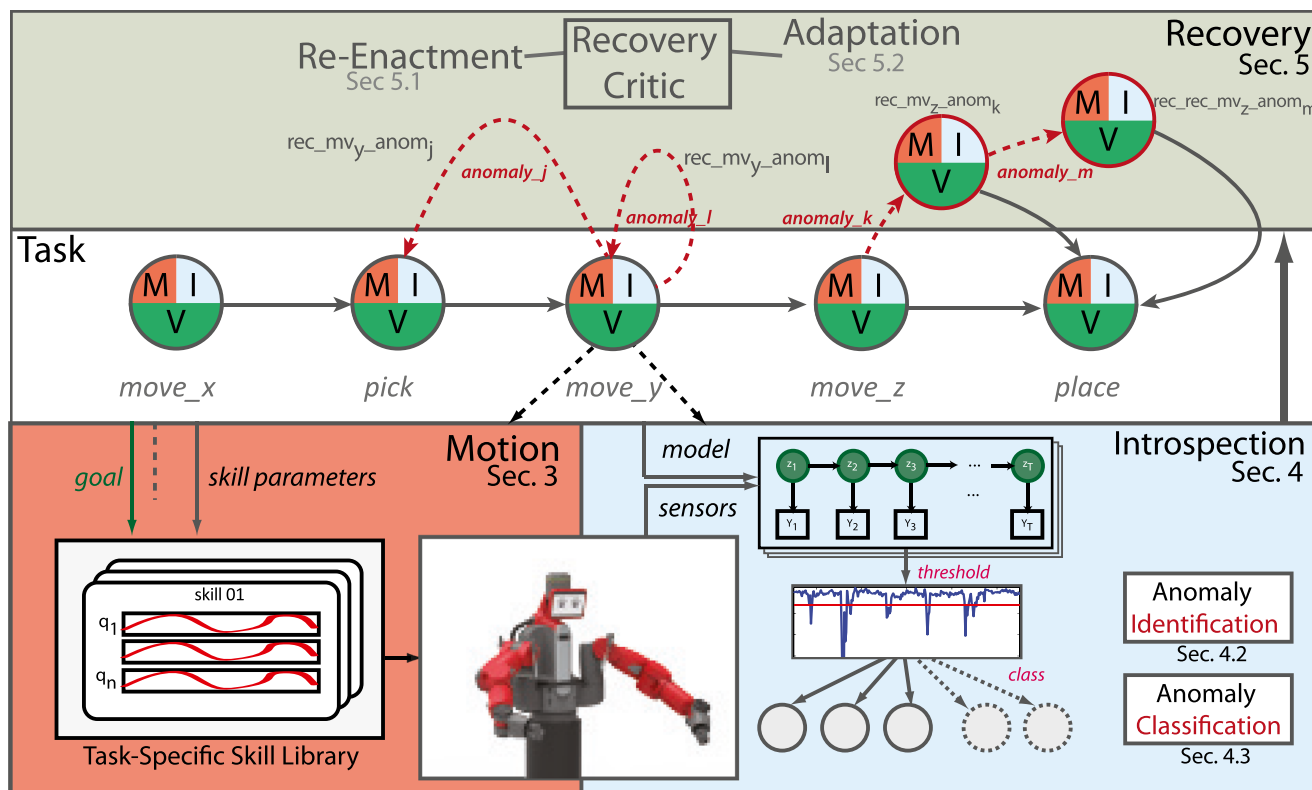


Fig. 2 Manipulation tasks are composed via directed graphs. Nodes contain three parallel running modules: motion, visualization, and introspection. Motion modules use pose goals provided by the visualization

module to generate desirable skills. Introspection modules use node-specific models to monitor for anomalies. If identified, classification ensues. A global critic then enacts a recovery policy

object (Fig. 4b) and transports it towards the box (Fig. 4c), after which, the robot appropriately places it in the box (Fig. 4d).

The kitting task is originally bootstrapped with 4 behaviors \mathcal{B} and 5 actions \mathcal{N} as shown in Fig. 3.

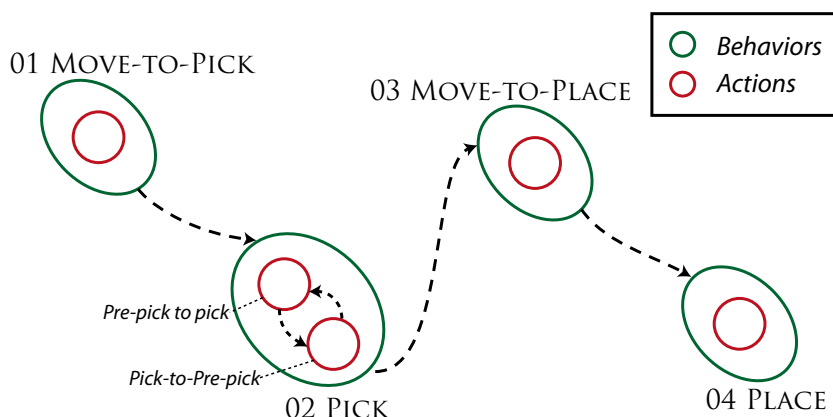
All behaviors except pick consist of single actions or nodes. The compound pick behavior consists of two nodes: “pre-pick to pick” and “pick to pre-pick”. The task requires that we train 5 actions and as such 5 skills, visualization goals, and introspection models. However, in the rest of

the paper, we will describe the task only in terms of the 4 high-level behaviors for simplicity. We will refer to them with the following codes for brevity: (i) MV2PK, (ii) PK, (iii) MV2PL, and (iv) PL.

2.1.2 External Disturbances

In this section we motivate the types of external disturbances possible in collaborative environments like a warehouse job as described above. Despite the interaction, we think

Fig. 3 Task graph for kitting experiment composed of 4 behaviors and 5 actions. Pick is a behavior composed of two actions. The task is thus constructed with 5 skills, goals, and introspection models. Modules are not shown explicitly in the node actions for clarity



that collaborative tasks (kitting here), can still result in low-cognitive demands for a human partner. The low-cognitive load might lead to monotony which in turn could lead to boredom and attention-loss. In such cases, a human co-worker may be more likely to accidentally collide with the robot or alter the environment in unexpected ways. A user may accidentally collide or unintentionally move a packaging object in ways the robot cannot model or anticipate as it tries to grip the objects. Object shifting (objects to be grasped or even the packaging box) may lead to tool-collisions, failed grasps, or even misgrasps. It is also possible that picked objects may slip from the robot's tool if the grasp is not optimal; or if upon motion, inertial forces acting on the object cause dynamics that break the grasp. Chains of anomalies are yet another possibility: human collisions that lead to object slips that move objects in such a way that lead to air grasps. As part of the discovered anomalies from Section 2.2, we introduce the basic anomaly types and their acronym in the interest of brevity: human collisions (HC), tool collisions (TC), object slips (OS), and no-object (NO). Section 4, will introduce the introspection methodology used to model robot skills including a description of our Anomaly Identification algorithms in Section 4.2) and Anomaly Classification algorithms in Section 4.3. Later, in Section 5 we introduce our recovery critic policies including Re-enactments (Section 5.1) and Adaptations (Section 5.2).

2.1.3 The Robot

A Baxter humanoid robot's right arm is used to pick commonplace objects set before him. The equipment used with the robot is: a 6 DoF Robotiq FT sensor, the standard Baxter electric pinching fingers, and Baxter's left hand camera. Each finger is further equipped with a multimodal tactile sensor composed of: (i) a four by seven taxel matrix that yield absolute pressure values, (ii) a dynamic sensor which provides a single capacitive reading in millivolts (mV) useful to detect tactile events, and (iii) an IMU and gyroscope [31]. Baxter's left hand camera is placed flexibly in a region that can capture objects in the collection bin with a resolution of 1280x800 at 1 fps (we are optimizing pose accuracy and lower computational complexity in the system) as seen in Fig. 4a. The use of the left hand camera facilitated calibration and object tracking accuracy. ROS Indigo on Linux 14.04 and a number of workstations are used to control all aspects of the experimentation. Code is available in our supplementary page [26].

2.1.4 Objects

A set of 6 common household objects consisting of box-like shapes and bottles were used in our work as shown

in Fig. 4a. The objects ranged in weight from 0.0308kg to 1.055kg and in volume from $3.2 \times 10^{-04} m^3$ to $1 \times 10^{-03} m^3$. The object's surfaces also varied slightly: some heavier objects had sleeker surfaces that incited object slips—we believe not an unreasonable determination as warehouses contain a wide variety of objects—whilst other objects had rougher surfaces. Across trials, object locations and order was varied to promote generalization.

Alvar tags, with 0.06m sides, were placed around the circumference of the objects for robust visual recognition (ALVAR can handle change in lighting conditions, optical flow-based tracking, and good performance for multi-tag scenarios) regardless of orientation (Fig. 4).

2.2 Cataloging Experiments

In this section, we describe the data collection process for the skill \mathcal{S} and introspection \mathcal{M} modules. Detailed presentations can be found in Appendix D & Section 4 respectively.

2.2.1 Motion Skill Training

In this work, DMPs were selected to encode motion skills.⁴ One-shot kinesthetic demonstrations were used to encode the five actions needed to bootstrap the kitting task.

2.2.2 Deducing Anomalies

As for the process of discovering what anomalies might exist in a given task, we hold that undeniably, robot researchers hold a bias towards which anomalies will exist and be discovered in a given task. To this end, we aim to discover the task anomalies by emulating a collaborative kitting task, including the human collaborator tedium and monotony that lead to unintentional changes or disturbances in the environment or the robot respectively.⁵

We tasked 5 robot researchers to act as the collaborative co-workers in the task under the monotonous conditions already mentioned (Section 2.1.2). Each user was trained to place the set of six household objects, one-at-a-time, in the collection area. From this exercise we extract two pieces of information: (a) anomaly classification labels (as judged by a human expert) that emerge from the task (see Section 2.1.2) and (b) the recording of all sensory-motor data. With respect to anomaly classification, we build anomaly models (Section 4) by recording $\pm 2secs$ of

⁴Out of expediency as the code was readily available. Using HMMs to generate motion would result in a more efficient architecture

⁵We did not expressly measure this, but hope to integrate it in future work.

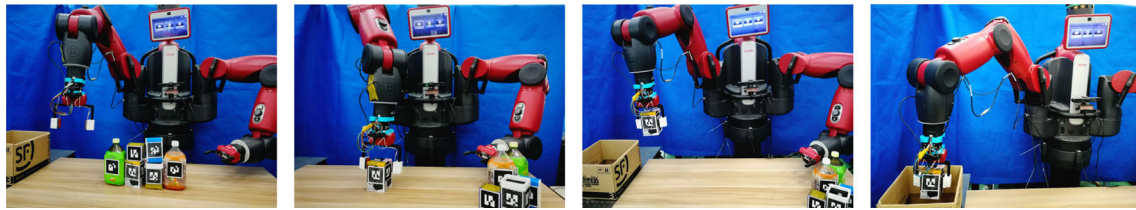


Fig. 4 Kitting experiment snapshots. Objects that need to be packaged are placed by a human collaborator before the robot in a collection bin. The shared workspace affords possibilities for accidental contact and unexpected alteration of the environment. The robot is tasked to pick-and-place each of the objects in a packaging box to its right. The visualization module uses the ALVAR tags to provide a consistent

multi-modal data around the time an anomaly is flagged.⁶ Note that in this work, we attempt to classify anomalies broadly. Consider a human collision; regardless of user the latter could occur at a high or low position; on the right or left side; its temporal occurrence or duration may differ significantly; as well as its intensity. Yet all of them fall under the same label. Our approach to classification is much broader than similar works [16] and renders the classification task more challenging.

2.2.3 Training and Inducing of Anomalies

Beyond the original data collection performed in Section 2.2.2, a second data collection round is conducted to improve training (parameter and hand-designed feature tuning). This round is performed iteratively seeking to maximize optimal performance while protecting against overfitting. The final number of training and testing trials used for anomaly identification and classification are described in Exp. 1 and Exp. 2 respectively.

2.2.4 Learning Recoveries

Upon the occurrence of accidental anomalies, re-enactments are learned from human users. Exp. 3 is used to learn probability models from human users given specific anomalies (Section 5.1.1). Similarly, for persistent anomalies, adaptations are incrementally trained through kinesthetic teaching. In Exp. 4, 5, and 6 a variety of adaptive skills are learned to address specific and emerging anomalies (Section 5.2).

3 Motor Skills and Learning from Demonstration

In this work, we use DMP to encode skills. DMPs encode dynamical systems through a set of nonlinear differential

equations with a point attractor system defined by a nonlinear forcing function that depends on a canonical system for temporal scaling. As DMPs are a well known framework, we do not present its details here, but do provide an overview in Appendix D for convenience.

The forcing term is set via kinesthetic teaching, where $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ with duration T are extracted as in [23]. The target forcing term is computed by rearranging (14) and integrating with the canonical system and substituting appropriate values to convert from time-mode to phase-mode.

Next, the goal is set to $g = x(T)$ and τ is selected such that a DMP reaches 95% convergence at $t = T$ before using standard linear regression to compute the weights w_i . Such procedure yields a baseline controller that can be improved by reinforcement learning [32] though this is not done in this work.

Motor skills are trained as individual *skills* (more robust methodologies [5, 23, 33] were not used here) for each phase of the task. Cartesian position and XYZ Euler representations are used to encode the attractor dynamics.

With respect to introspection models, we leverage sensory-motor signatures to learn the structure of sensory responses to motion data [22, 34]. Our observations consist of a 6 DoF end-effector twist and wrench respectively, a 7 DoF pose (using quaternions as orientation), and 56 tactile values (each finger has 4-by-7 taxels). All observations were hand-processed into features as detailed in Appendix E. All object poses are acquired using AR codes through the ROS ALVAR framework.⁷

We demonstrate five simple skills: move-to-pick, pre-pick-to-pick, pick-to-pre-pick, move-to-box, and place. We ensure that skills are executed in such a way that no occlusion occurs. Skills are executed at least 7 times to obtain sensor information of nominal skills to train introspection model (Section 4). Once DMP and introspection models are trained, they are stored in their corresponding libraries. Then, a behavior graph is constructed where nodes contain

⁶An online Redis database is used as rosbags can only be processed offline.

⁷http://wiki.ros.org/ar_track_alvar.

appropriate ID types that are handled by the system to enact necessary models during task execution. As for transitions, *nominal nodes* currently transition to only one other node, so no explicit transition classification is enacted. For anomalies however, transitions to different nodes will depend on the anomaly classification (Section 4.3) and the selected recovery policy (Section 5).

3.1 Goal Setting

For task execution, the Visualization module is responsible for selecting appropriate goals for enacted skills. Currently, goals are pre-specified according to the nature of the skill. The pre-pick node uses the pose obtained from the Alvar code placed on objects (we queue multiple poses from right-to-left order). The pick node uses the actively tracked object’s pose as a goal. The move-to-box node uses the centroid Cartesian location of the box from the top-view. For the place skill, we pre-assign object locations to different portions of the packaging box according to the total number of objects (we assume the box is sufficient to pack all objects in one dimension). In the future work, we aim to intelligently assign goals by exploiting affordances in the task.

Additionally, we highlight that though the skill set used in this work is simple, the space of possible anomalies is significant and is this work’s main focus. To this end, in our experimentation, we test strenuous anomalous conditions that could emerge in unstructured environments. (Section 6).

4 Robot Introspection

Robot introspection is a precursor to policy recovery. In this work, we use previous developments with a non-parametric Bayesian Markov Jump Linear Systems (MJLS) system for anomaly identification [17] and classification [25] albeit with an improved variational inference algorithm. This

section will first provide an overview for the Bayesian non-parametric model and then present more specifics for anomaly identification and classification. Figure 5 summarizes the introspection system flow.

4.1 Bayesian Non-parametric Hidden Markov Modeling

Robot introspection uses Bayesian non-parametric Markov Jump Linear Systems (MJLS) and memoized variational inference with scalable adaptation as the modeling mechanism. A non-parametric Bayesian HMM, namely the sticky Hierarchical Dirichlet Process (sHDP) HMM can be used to learn a VAR process (sHDP-VAR-HMM). Such an approach enables us to both learn the model complexity (number of latent states) directly from the data and model mode-specific observations through linear dynamics without independence assumptions [35].

Recent advances in variational inference allow to process large datasets incrementally and optimize the creation and removal of states yielding highly optimized models that are simpler, more compact, more interpretable, and better aligned to ground truth state segmentations [36].

In this section we provide basic overviews of HMMs, the sHDP-VAR-HMM, and the variational inference mechanism for completeness. For further details, please consult the appropriate references.

4.1.1 Hidden Markov Models

HMMs are a doubly stochastic and generative process used to make inference on temporal data. The underlying stochastic process contains a finite and fixed number of latent states or modes z_t which generate observations $\mathcal{X} = \{x_t\}_{t=1}^N$ through mode-specific emission distributions $b(z_t)$. These modes are not directly observable and represent sub-skills in a given task node. Transition distributions, encoded in transition matrix π_{ji} , control the probability of transitioning across modes over time. Given the initial mode

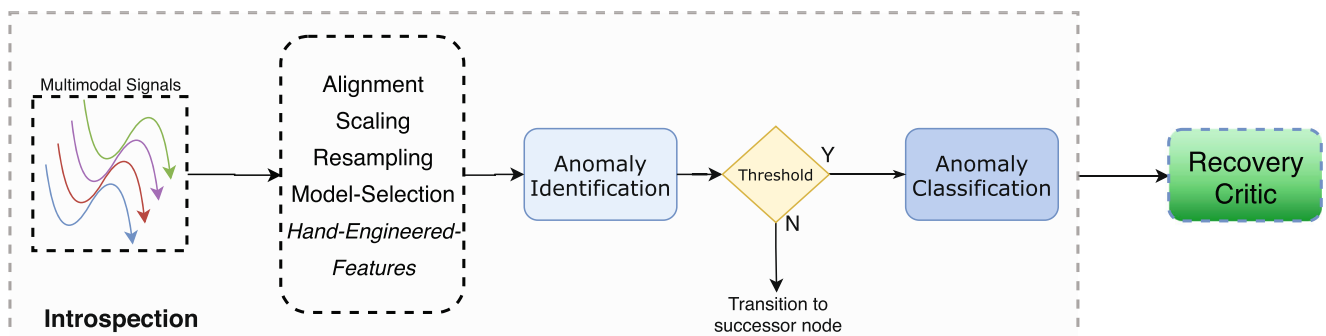


Fig. 5 The robot introspection module flow diagram. A continuous anomaly detector flags and classifies anomalies when the sensory-motor signals deviate from the expectation (Section 4). After introspection, control is directed to the recovery critic (Section 5)

distribution π_0 and a set of observations, the Baum-Welch algorithm is used to infer model parameters $\Pi = (\pi, b)$. HMMs assume a fixed number of latent states as well as mode-specific conditionally independent observations. Such assumptions limit the expressive power of HMMs as they are unable to derive natural groupings and model complex dynamical phenomena.

4.1.2 The sHDP-VAR-HMM

Bayesian non-parametric priors extend HMM models to learn latent complexity from data [25, 35–37]. As such, we leverage the non-parametric priors to learn a better complexity for the latent modes and their transition distribution. To allow for a flexible number of latent states, the stochastic Dirichlet processes (DP) G is used. DPs enjoy an unbounded number of θ_k support points that help with modeling whatever complexity the data churns out.

Instead of using a traditional transition distribution over the latent states $z_t|z_{t-1} \sim \pi_{z_{t-1}}$, a set of transient *probability measures* G_j can be used. Equation 1 defines a DP with an infinite number of independent probability measures θ_k to model transition (G_{jk} with probability mass π_{jk}).

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k} \equiv DP(\alpha, G_0) \tag{1}$$

The DP uses a base distribution G_0 to control the distribution of the probability measures and is chosen to allow for computational efficiency via conjugate analysis. The probability mass is set via the Griffiths-Engen-McCloskey $GEM(\gamma)$ distribution (otherwise known as the stick-breaking process), which is controlled by the α parameter.

The next step requires us to ensure that each of the indexed transition probability measures G_{jk} share the same support points as the process is sampled. As the DP is a stochastic process, sampled subsets will result in entirely different support points. To avoid this, a hierarchical DP construction is necessary where the base measure yields a fixed global set of support points for the parent distribution. The base measure would then be defined as $G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$, where $\beta|\gamma \sim GEM(\gamma)$.

The HDP-HMM, in this form, does not yet differentiate self-transitions from moves between distinct latent states and allows for fast switching dynamics between them and causing significant posterior uncertainty. For this reason, a “sticky” self-transition bias parameter is introduced that favors self-transitions as described in [35].

For the observational model of the sHDP-HMM we use a VAR process that is useful to model complex phenomena. Instead of independent observations, each

mode is characterized by conditionally linear dynamics of the past r mode-dependent observations as seen in Eq. 2.

$$x_t = A_k x_{t-1} + e_t(z_t = k) \tag{2}$$

Where, we consider the first-order ($r = 1$) auto-regressive Gaussian likelihoods with additive white noise e and observation x_t . Note that each state k is composed of a time-invariant regression matrix coefficients A and a covariance matrix Σ .

A and Σ , are both uncertain and need to be learned. The parameters $\theta = \{A, \Sigma\}$ are approximated for each state by defining a conjugate prior distribution on them. Particularly, a Matrix Normal Inverse Wishart (MNIW) is used as a conjugate prior distribution when both A and Σ are uncertain. If only the covariance is uncertain, the conjugate prior is defined as d -dimensional Inverse Wishart (IW) distribution with covariance parameter Δ , a symmetric positive definite scale matrix and ν the degrees of freedom as in Eq. 3.

$$\Sigma \sim \mathcal{IW}(\nu, \Delta) \tag{3}$$

The full definition of this joint prior is found in [36] and defined as $\mathcal{N}\mathcal{IW}(\kappa, \vartheta, \nu, \Delta)$. For the *IW*, the first moment of the distribution is:

$$\mathbb{E}[\Sigma] = \frac{\nu \Delta}{\nu - d - 1} \tag{4}$$

where, ν , is the degrees of freedom. The expectation of the covariance, for N exemplars of data \mathcal{X}_N for a given skill and a sequence with length T_n , is defined as:

$$\mathbb{E}[\Sigma] = s_F \sum_{n=1}^N \sum_{t=1}^{T_n} (x_t - \bar{x})(x_t - \bar{x})^T \tag{5}$$

Then, to determine the matrix A of regression coefficients, we use the matrix-normal inverse wishart (MNIW) distribution, which places a conditionally matrix-normal prior on A (for a given latent state) such that:

$$A|\Sigma \sim \mathcal{MN}(A; M, \Sigma, K) \tag{6}$$

The matrix normal is computed once Σ is available, where the covariance Σ represents the covariance across the rows, while K represents the covariance across the columns.

By using the model over a set of multi-modal exemplar data \mathcal{X}_n , the sHDP-AR-HMM can discover and model shared behaviors in the anomaly data across exemplars, even from a few examples. This model does assume however that all exemplars share the same (latent) modes and that modes switch amongst themselves in the same way). It is also possible to use a beta-process prior [35] to avoid this limitation, but this has not yet been implemented for online performance. Pseudo-code for the generation of skill models using the sHDP-VAR-HMM is outlined in Algorithm 1.

Algorithm 1: sHDP-VAR-HMM models for classification.

```

Input:
 $N_c$ : Number of sequences for class  $c \in \mathcal{C}$ ;
 $\{\mathcal{X}_n\}_{n=1}^{N_c}$ : Dataset with  $N_c$  sequences, each of length  $T_c$ ;
 $N_i$ : Number of the maximum iteration for learning;
 $N_r$ : Number of runs for the whole learning;
 $random\_state$ : The random number generator;
 $k\_splits$ : Number of folds;
 $a, b, d, e$ : Hyper-prior for concentration parameters;
 $v, \Delta, V, M, s_F$ : Hyper-prior for the MNIW distribution;
 $\kappa$ : The self-transition bias;
 $K$ : The truncation active states.
Result: HDP-HMM models for each class
 $\{\mathcal{X}_n\}_{train}, \{\mathcal{X}_n\}_{test} =$ 
KFold_split( $\{\mathcal{X}_n\}_n^{N_c}, k\_splits, random\_state$ )
for  $k$  in  $k\_splits$  do
  for  $i$  in  $N_r$  do
    for  $n$  in  $N_i$  do
      if not converged then
         $\Theta_\pi, loss = \text{HDPHMM}(\{\mathcal{X}_n\}_{train}, a, b,$ 
         $d, e, v, \Delta, V, M, s_F, \kappa, K)$ 
      else
        | HDP-HMM with  $\Theta_\pi$  and  $loss$ 
      end
    end
     $\{loss\}_i^{N_r} \leftarrow loss$ 
    if  $i == N_r$  then
      |  $\Theta_\pi \leftarrow$  with the minimum loss value
    end
  end
   $\mathcal{L}_{test\_mean} = \frac{\sum_{i=1}^{N_{test}} p(\{\mathcal{X}_n\}_i | \Theta_\pi)}{N_{test}}, i \in N_{test}$ 
   $\{\mathcal{L}_{test\_mean}\}_k^{k\_splits} \leftarrow \mathcal{L}_{test\_mean}$ 
  if  $k == k\_splits$  then
    | return  $\Theta_\pi$  with the maximum  $\mathcal{L}_{test\_mean}$ 
  end

```

4.1.3 Memoized Variational Inference with Scalable Adaptation

Prior to the work in [36], inference algorithms for HMMs and HDP-HMMs had not efficiently learned from large datasets nor have they effectively explored data segmentations with varying number of states. Inference algorithms can be trapped at local optima near their initialization points. Stochastic optimization methods, which are unable to update the number of modes after execution, are particularly vulnerable to data segmentation and exploration and local optima [38, 39]. These methods may yield states that

become irrelevant and should be removed. Recently, algorithms that add and remove states via split and merge moves have been designed for non-parametric priors like HDP and BP algorithms [37, 40]. However, these Monte Carlo proposals suffer from scalability as they must use the entire dataset and also require that all sequences fit in memory.

Hughe’s et al. memoized variational inference algorithm with scalable adaptation uses birth proposals to create new states and merge and delete moves to remove poor predicting states; however, adaptations are validated through a global variational bound [36]. The algorithm caches sufficient statistics and parallelizes local inference steps to efficiently process sequence subsets at each time step to allow for rapid adaptation of the state space cardinality. The inference algorithm outputs all around better models—more compact and interpretable—to infer the sHDP-HMM’s posterior distribution leading to better classification results. We use it in our models for the first time in this work. For further details, see [36] and [41] for the open-source code.

4.2 Anomaly Identification

Anomaly identification continuously monitors robot behavior to identify unexpected behaviors during skill execution and also *during recovery phases*. Recovery phases are challenging as they usually begin in anomalous states and are more likely to trigger false-positives [20]. Different metrics for anomaly identification have been suggested in the literature [15, 18, 20, 25]. Most of these techniques use the maximum cumulative log-likelihood value of the observations given a model. In [17], it was shown that such metrics performance is limited during recovery stages as the standard deviations for the early steps are very narrow.

In [17], we presented a metric based on the natural logarithm of the HMM filtered belief state (from hereon referred to as the “forward gradient” measure) ∇L . Given an HMM model Π and an incoming time series $x_{1:t}$, the natural logarithm of the filtered belief state (see 17.4.1 [42]) associated with the forward model for latent state i can be represented according to Eq. 7.

$$L_t = \log \sum_{i=1}^N \alpha_i(t) = \log \sum_{i=1}^N \exp(\log \alpha_i(t)). \tag{7}$$

The forward term can be computed iteratively from the previous time-step result. We later established that the forward gradient L -curve depends on the latest emission probability of the HMM model, which in-turn depends on the latest observation. The key point is the stable and large positive-valued gradients are generated when observations are generated by the true latent state, hence giving us a way to perform skill identification.

From this fact, derived an anomaly detection heuristic. We compute the max, min, and gradient range value

between these and ultimately set the per-mode threshold to:

$$\nabla L_t^m(Y) < \nabla L_{min} - \frac{\nabla L_{range}}{2}. \tag{8}$$

The metric was shown to yield accurate, robust (precision and recall), and fast anomaly identification, even in recovery stages. Fig. 6 illustrates the identification performance of the forward gradient approach. We describe parameters values, models, and training and testing details in Section 6, whilst anomaly Identification results are found in Exp. 1.

4.3 Anomaly Classification

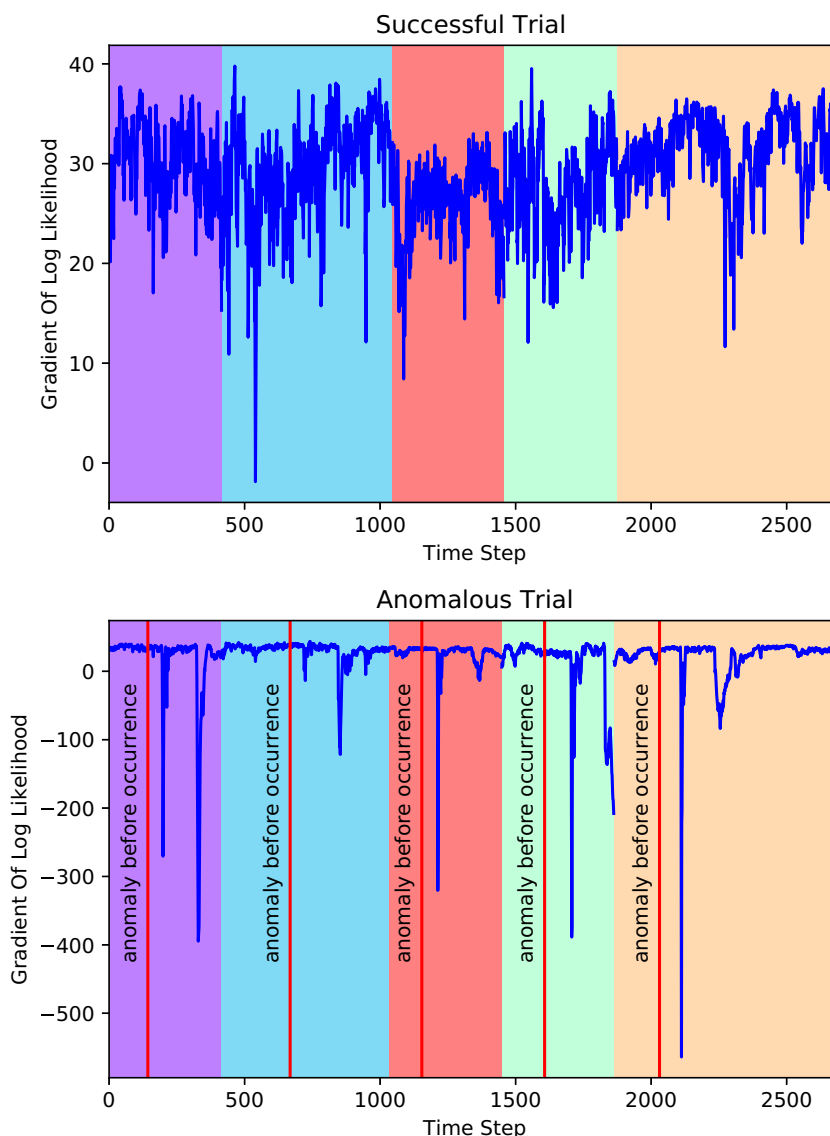
The anomaly classification service is triggered once an anomaly is identified.

We classify anomalies caused by external disturbances generated either by intrusive human behavior or resulting

from poor modeling or anticipatory ability on the robot’s end. As introduced in Section 2.1.2, five anomaly classes emerged in the cataloging experiments of the kitting task: (accidental) human-collisions (HC) in a shared-workplace; wall-collisions (WC) with the packaging box; tool collisions (TC) when moving to pick or place; object slips (OS) caused by inertia or external disturbances; and the unexpected movement of objects that led to missed grasps and described as “No Object” (NO).

Just as with anomaly identification, the sHDP-VAR-HMM was used. Given M trained models for M robot skills, 3-fold cross validation is used along with the standard forward-backward algorithm to compute the *expected cumulative likelihood* of a sequence of observations (within the sliding window) as: $\mathbb{E}[\log P(\mathcal{X}_i | \Pi_m)]$ for each trained model $m \in M$. Given a test trial x , the cumulative log-likelihood is computed given the

Fig. 6 The log-likelihood gradient ∇L for 5 motor skills s in a task \mathcal{B}_e . Top plot shows a nominal task whose ∇L is steadily positive. Bottom plot shows one anomaly per skill execution. Anomalies occurred shortly after the red vertical lines seen in each skill. When an anomaly occurs, the gradient becomes distinctively negative



observations conditioned on all available trained skill model parameters $\log P(x_{m_1:m_t} | \Pi_m^M)$ at a rate of 200Hz. The process is repeated when a new skill is started. Given the phase in the manipulation graph m_c , we can index the correct log-likelihood $\mathbb{I}(\Pi_m = m_c)$ and see if its probability density of the test trial given the correct model is greater than the rest for the last observation point:

$$\log P(x_{m_1:m_t} | \Pi_{correct}) > \log P(x_{m_1:m_t} | \Pi_m), \quad (9)$$

$$\forall m(m \in M \wedge m \neq m_c).$$

Further information regarding, parameters values, models, and training and testing are presented in Section 6. Anomaly classification results are detailed in Exp. 2.

5 Anomaly Recovery

After classification, the recovery critic implements recovery through re-enacting or adaptive policies as shown in Fig. 2. Re-enacting policies re-execute a skill (possibly the current skill or a previous skill) as designated by the policy (Section 5.1). Adaptive policies resolve persistent errors by training adaptive skills that leverage human understanding into the complex set of world-object-robot relations (see Section 5.2). The recovery critic runs, not only during all normal phases of the task, but also significantly, during recoveries of anomalous events. To illustrate, refer to Fig. 2, where it is seen that for node *move_z*, a persistent anomaly *anomaly_k* led to the creation of an adaptive skill found in node *rec_mv_z_anom_k*. Then, during the execution of this adaptive skill, a new persistent anomaly *anomaly_m*

entered the system. Our framework identifies it and assigns a new adaptation encoded in node *rec_rec_mv_z_anom_k* that enables the system to reach the next milestone.

Implemented recoveries, whether re-enacting or adaptive, are strictly coupled to the specific anomalies that caused them. Recoveries themselves are globally unique and thus emerge contextually in the task (not so with anomalies). That is, the same anomaly may occur at different points in a task, e.g. a tool collision may happen as we try to pick an object; as we move to the packaging box; or as we place the object in the box. However, the recoveries associated with these anomalies are unique. An overview of the recovery framework is summarized in Fig. 7.

5.1 Re-enacting Policies

Re-Enacting policies resolve accidental one-off anomalies. All anomalies are considered accidental by default and only when they cannot be resolved through re-enactment are they considered persistent. The premise is that accidental events are resolved through the re-enactment of re-parameterized skills. The key question is to identify which skill needs be re-enacted. A few works have used a policy where either the entire task is repeated from the beginning or fixed points in the task are selected a priori [19, 20, 43]. In this work, we learn more efficient skill selection mechanisms.

Given a current milestone \mathcal{N}_j , for each new accidental anomaly \mathcal{F}_y , a new re-enactment (transition) \mathcal{R}_R is inserted into the graph as follows:

$$\mathcal{R}_R : \mathcal{T}_{\mathcal{N}_j, \mathcal{N}_*} | \mathcal{F}_y, \mathcal{N}_j \quad (10)$$

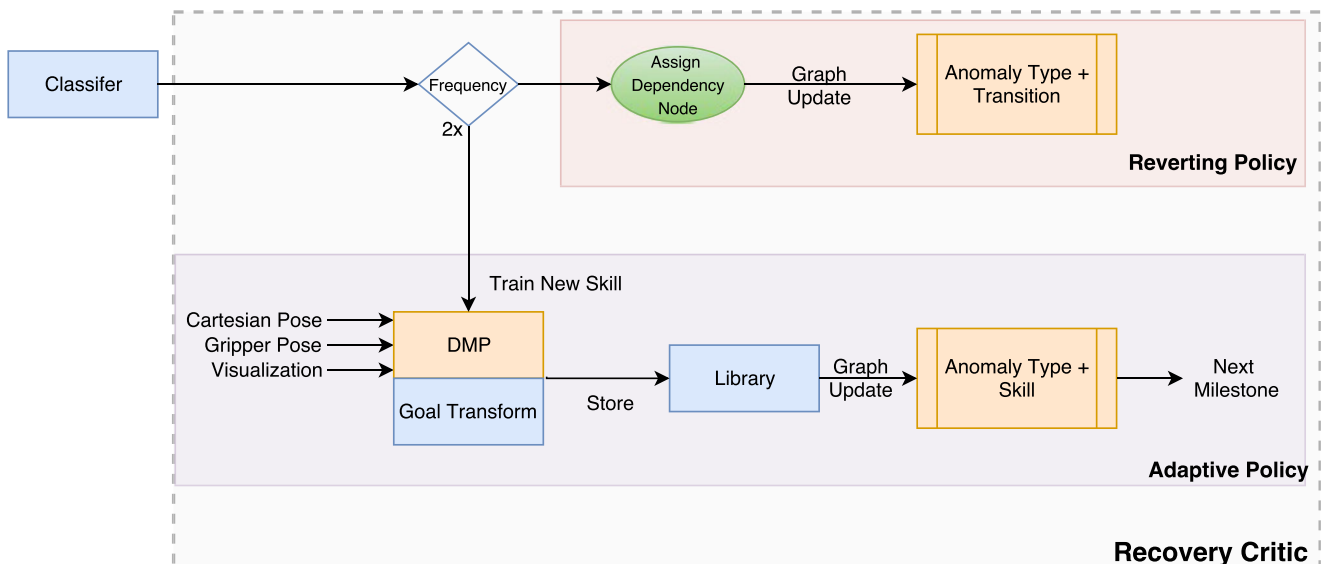


Fig. 7 After classification, the recovery critic triggers a re-enacting or an adaptive policy according to the nature of the anomaly: persistent or accidental. Re-enacting policies model human decision making

probabilistically (Section 5.1). Adaptive policies train a new skill and transform the goal to reach a next phase in the task

where * is the target node and it is selected according to the policy introduced in Section 5.1.1. To illustrate, consider an OS anomaly during node 3 when the robot is moving towards the box. Instead of returning home, the robot can re-enact a re-parameterized version of the pick skill.

Provided built-in safety procedures, once the temporary accidental contact concludes, the robot could re-enact the current skill. Notice that nodes contain skills that are inherently reactive. The starting and goal poses of a skill can be set without altering the skill's properties. A re-enactment of the current skill with a re-parameterized starting pose would be enough to complete that task phase and reach the next milestone.

In effect, re-enactment goal nodes are chosen **in relation to the nature of the anomaly type**.

5.1.1 The Re-enactment Policy

Re-enactment goal nodes are assigned through multinomial distributions that model human-user goal node selections given a current node and a specific anomaly. Five human users studying a robotics master's degree were trained to understand the graph topology of the task, possible transitions, skill execution, goal parameterization, anomaly types, and legal node selections/transitions for re-enactment. Each user examined 5 trials of induced anomalies on a per-node, per-anomaly basis, yielding independent multinomial distributions to determine re-enactment policies. For instance, if at node 2, three anomaly types occur, then there will be three multinomial distributions modeling the policy. For each multinomial, let $N = (N_1, \dots, N_K)$ be a random vector where N_j is the number of times a node j is selected as a re-enactment target node. Then N has the following pmf:

$$\text{Mu}(N|n, \theta) = \binom{n}{N_1 \dots N_K} \prod_{j=1}^K \theta_j^{N_j} \quad (11)$$

where, θ_j is the probability that node N_j is selected. Given the node sequence of Fig. 3, the results are shown in Table 1. The multinomial provides an indirect way to represent human intuition about the complex set of relations that exist between the robot (and its limbs), the relevant objects of the task at hand, and the interactions that the robot and the objects have with the world. Additionally, the multinomial also encode a person's internal belief about the utility of a choice, his/her own learning ability (within a trial and across trials), and the person's risk propensity or aversion in decision making.⁸ For instance, OSs that

Table 1 Human user selections of re-enactment target nodes given a current node execution and a specific anomaly

Anomaly	Source Node	Target Node	Count
HC	MV2PK	MV2PK	25
	PK	PK	25
	MV2PL	MV2PL	25
	PL	PL	25
TC	PK	MV2PK	25
	MV2PL	MV2PL	5
OS	PK	PK	20
	PK	MV2PK	5
	MV2PL	PK	25
NO	PK	PK	24
	PK	MV2PK	1

occurred during the picking skill (node 2), were assigned two different types of re-enactment target nodes: to re-execute the same pick skill with 80% probability and to execute the previous move-to-pick (node 1) with 20% probability. The choice of returning to node 1 represents a more conservative belief or risk averse selection on the user's part.

5.1.2 Re-Enactment Target Nodes

Goals for re-enacted target nodes are set by the visualization module. The starting pose is simply the current pose at the time of anomaly, while the goal pose is set as originally described in Section 3.1.

5.1.3 Training Re-Enactments

Re-enactment policies designed in this section were trained during the cataloging experiments of Section 2.2. Success rates for re-enactment policies given accidental anomalies are reported in Exp. 3-6 under a variety of different conditions.

5.2 Adaptive Policies

Adaptive policies are used to resolve persistent anomalies. Persistent anomalies are classified as such when a re-enactment policy fails to resolve a given anomaly twice consecutively. This phenomena indicates that re-enactment is unable to solve the condition and that the task requires explicit adjustments to finish the task successfully. In this work, we rely on human intuition and expertise to provide the necessary adaptation skill to solve the persistent task anomaly.

⁸Utility theory seeks to model complex decision-making processes using notions of expected value, expected utility, risk and learning models, and more [44]

5.2.1 Kinesthetic Teaching

Our system is designed to pause automatically when two consecutive re-enactment policies occur for the same node-anomaly pair in the graph. The system then awaits for the user to initiate kinesthetic teaching (through the push of a system button) and encode the adaptive skill. The system also, at this time, records all relevant sensory-motor data necessary (until the end of kinesthetic teaching) to train a new introspection model for the current nominal (adaptive) skill.

5.2.2 Graph Integration

Given a current milestone \mathcal{N}_i , for each new persistent anomaly \mathcal{F}_x , a new adaptive recovery node $\mathcal{R}_A : \mathcal{N}_{ij}$ is inserted into the graph as a new branch in-between milestones. The target node \mathcal{N}_j is inherited from the parent resulting in the need an additional transition to the successor milestone in accordance to Eq. 12:

$$\mathcal{R}_A : S_{\mathcal{N}_i} | \mathcal{F}_x \rightarrow \mathcal{N}_{ij} \tag{12}$$

$$T_{\mathcal{N}_i, \mathcal{N}_{ij}, \mathcal{N}_j}$$

Figure 2 illustrates the concept, consider how in node *move_z*, persistent anomaly *anomaly_k* is resolved using adaptive skill *rec_mv_z_anom_k* as a new branch between milestones *move_z* and *place*.

For cases in which an anomaly \mathcal{F}_{xx} occurs during an adaptive node (recovery-over-recovery) \mathcal{N}_{ij} , a new adaptive node is created in a new branching layer, whose target is still the successor milestone:

$$\mathcal{R}_A : S_{\mathcal{N}_{ij}} | \mathcal{F}_{xx} \rightarrow \mathcal{N}_{ijk} \tag{13}$$

$$T_{\mathcal{N}_{ij}, \mathcal{N}_{ijk}, \mathcal{N}_j}$$

Adaptive nodes always transition to the ensuing milestone, no matter the branching level. In this work, we have assumed that a single adaptive skill is sufficient to restore the nominal functioning of the task. It is plausible to sequence skills to achieve more complex manipulations.

5.2.3 Setting Adaptive Node Goals

Skill goals are set by the Visualization module of a node (Section 3.1). However, for adaptations, when human users introduce additional manipulation, they are also introducing a transformation on the goal pose of the parent skill with respect to the base frame. Adaptive skills then compute the transformation of the last time step in kinesthetic teaching with respect to the goal of the parent node. During online testing, the Visualization module computes the real-time goal of the parent node, whilst the adaptive skill transforms that goal to achieve task generalization during adaptation.

5.2.4 Inheriting Re-Enactment Policies

Whenever we push a new adaptive node into the graph, that adaptive node is set to inherit the same re-enactment policies available to its predecessor. This is important so as to avoid the need to re-train re-enactments in new adaptation nodes.

5.2.5 Training

Cataloging experiments were used to capture sufficient data to create robust nominal skill introspection models for adaptive anomalies. These models are then used by our Anomaly identification routine in Section 4.2, to identify anomalies that may occur during such adaptations. Anomaly Identification performance is presented in Exp. 1, whilst the success rates for adaptive policies presented in this section are reported in Exp. 4-6 under a variety of different conditions that elucidate system performance.

6 Experiments and Results

Seven experiments are setup to test the efficacy and versatility of our recovery policies under different situations. Nonetheless, as recovery is inextricably married to the performance of the anomaly identification and classification systems, it is imperative that a comprehensive evaluation on these subsystems is also carried out. Our previous works have analyzed the accuracy, robustness, and reactivity of anomaly identification and anomaly classification in their own context. This work expands their scope with a more extensive number of classes and situations induced by the more unstructured collaborative environment. Exp. 1-2, present accuracy and robustness results for Anomaly Identification and Anomaly Classification respectively. Exp. 3-6 examine the recovery policy efficacy and versatility. Exp. 3 measures the robustness of re-enacting recovery policies. Exp. 4 tests the robustness of adaptation policies. Exp. 5 analyzes the robustness of the co-existing policies. Exp. 6 tests the system’s ability to introspect and recover from anomalies that occur during an ongoing recovery action. Finally, Exp. 7 analyzes the reactivity of our system. Experimental descriptions are presented first, followed by the results and their analysis.

6.1 Kitting Setup

As stated in Section 2.1, the Baxter robot is set-up to perform a kitting experiment in conjunction with a human co-worker. The human places objects in the collection bin and the robot packs them. The shared space makes is susceptible to human accidental and persistent anomalies.

Three computers are used to run the experiment: Baxter's internal computer, which runs Gentoo Linux 2.2 and an Intel(R) Core i7-3770 CPU@3.40GHz, 4GB-RAM, x64-based processor. The internal computer is used to run a ROS joint trajectory server as well as the camera on the left arm. The other two computers run Linux 14.04 with ROS Indigo. One computer has an Intel(R) Core i5-3470 CPU@3.20GHZ, 6GB-RAM, x64-based processor and runs alvar recognition, the moveit service, and time-series pre-processing for all sensory-motor data. The second workstation, runs an Intel Xeon i7-6820HQ CPU@2.70GHZ (3.60GHz Turbo), 8MB-RMA, x64-based processor and is in charge of running online anomaly identification and anomaly classification which is implemented with BNPY [41] under a ROS-wrapper.

Our graph implementation uses a hybrid approach. Base nodes for the kitting experiment are currently implemented through ROS-SMACH. The non-adaptive nodes however are designed through an internal procedural representation detailed in Appendix A [26]. Diagrammatic representations and code are accessible through our supplementary materials page [26].

Additional experimental details like human testing are presented in Appendix E, while signal pre-processing details are presented in Appendix E.

6.2 sHDP-AR-HMM Parameters & Hyperparameters

Given that both anomaly identification and classification are based on the same model, we use a base-model to introduce parameter settings that are broadly shared across the methodologies. Whenever particular differences exist from the base-model, they are explained within specific experiments. Parameter settings are presented through Appendix D.

6.3 Classification Modalities

As part of Exp. 3-6, we present success rate metrics as a function of two distinct classification system modalities:

- i. perfect anomaly classification (independent system)
- ii. imperfect classification (combined system)

The **perfect anomaly classification** modality only attempts recoveries when *true positives classifications* are produced. This modality treats the entire system as three (anomaly identification (AD), anomaly classification (AC), and the recovery (REC)) independent sub-systems. By separating the sub-systems we can study their effectiveness independently from the other systems. The **imperfect classification** modality considers recovery success rates in the presence of misclassifications. This treats the system two (AD and AC/REC) dependent subsystems. The

separation allows us to study some interesting phenomena that emerged from the REC system and is detailed in each of the experiments.

6.4 Experiments

A description of all experiments is presented before presenting the results. Note that our supplemental video [26] depicts many of the cases described herein.

6.4.1 Experiment 1. Anomaly Identification

In Exp. 1, we evaluate the performance of the anomaly identification system across the entire set of experiments.

Since the anomaly identification system is the first to be triggered, it is critical that identification is done accurately; otherwise the system will suffer from upstream errors. In this section we present the system's identification accuracy, precision, and recall metrics.

The anomaly identification system used the sHDP-VAR-HMM technique (Section 4.1) to create class models for the nominal (non-adaptive) skills of Section 2.1 and the adaptive skills (that resolve persistent anomalies) of Exp. 4a,b,c, and 6a,b. Once the nominal models are trained, the forward gradient measure (Section 4.2) is used for anomaly identification. Upon the collection of training (offline) data from the induction experiments (Section 2.2), a scoring heuristic was implemented over 5-fold cross-validation that allowed us to select from a variety of hand-engineered multi-modal signal features and parameter values. Different combination of features were tested for specific sets of parameter values (see Appendix D and [26]). Scoring in the form of accuracy, precision, and recall metrics are computed for each combination as show in Fig. 8.

The highest scoring model was selected and resulted in the following combination:

Feat. Comb	#3	#4	#5	#7	#8	#9	Avg
1	0.959	0.836	0.748	0.506	1.000	0.889	0.823
2	0.329	0.343	0.315	0.012	0.000	0.000	0.167
3	0.986	0.022	0.162	0.012	0.000	0.000	0.197
4	0.000	0.000	0.712	0.639	0.692	0.000	0.341
5	0.973	0.970	0.748	0.807	0.231	0.000	0.622
6	0.952	0.993	0.243	0.398	0.846	0.444	0.646
7	0.986	0.993	0.757	0.217	0.923	1.000	0.813
8	0.993	1.000	0.802	0.867	0.231	1.000	0.816
9	0.959	0.948	0.757	0.904	0.846	1.000	0.902
10	0.966	0.985	0.757	0.855	0.923	1.000	0.914
11	0.973	0.888	0.775	0.855	1.000	1.000	0.915
12	0.973	0.993	0.820	0.720	1.000	1.000	0.918
13	0.979	1.000	0.775	0.855	0.923	1.000	0.922
14	0.986	1.000	0.793	0.759	1.000	1.000	0.923
15	0.965	0.763	0.750	0.975	0.923	0.556	0.822

Fig. 8 Anomaly identification accuracy for a distribution of features across nodes. Low, medium, and high probabilities are show in red, blue, and yellow respectively

- End-effector force F , torque τ , linear velocity v , and angular velocity ω such that: $[F, \tau, v, \omega] \in \mathbb{R}^3$.
- l^2 -norm of the above signals; namely: $[|F|, |\tau|, |v|, |\omega|] \in \mathbb{R}^1$.
- The maximum standard deviation σ computed for each of the 28 taxels in a tactile sensor for the left and right fingers; namely, $\max_{\sigma} [\sigma_l, \sigma_r] \in \mathbb{R}^1$.

To build anomaly identification models for both non-adaptive and adaptive skills, a fixed number of 7 trials was used. Non-adaptive skills consisted of the move-to-pick, pick, move-to-box, and place skills and adaptive skills are those captured in Exp. 4a,b,c and 6a,b respectively.

Macro accuracy, precision, and recall metrics are extracted by testing whether we can identify anomalies (HCs, TCs, OSs, NOs, or WCs) given some domain (nodes or sub-experiments).

6.4.2 Experiment 2. Anomaly Classification

After anomaly identification, it is important to understand the performance and robustness of the anomaly classifier. The anomaly classification also uses the sHDP-VAR-HMM with memoized variational inference (Section 4.1) along with the same features and training style used in anomaly identification. The model is trained to classify anomalies caused by human collisions (HCs), tool collisions (TCs), object slips (OSs), no objects (NOs), and wall collisions (WCs) introduced in Exp. 4. For training, we used the following number of trials for the aforementioned classes: HC-18, TC-17, OS-18, NO-15, and WC-17. We have not yet implemented an unsupervised learning method that automatically generate new anomaly labels based on previously unseen data (determined through a confidence metric), but we have contemplated this work (see Section 7). Anomaly classification is only triggered if anomaly identification experiences a true-positive. Once the classification procedure is called no true-negatives or false-negatives exist in the system. Only true or false positives. For this reason, classification will be measured in terms of accuracy across nodes or confusion matrices for Exp. 3-6.

6.4.3 Experiment 3. Testing Re-enactment

Experiment 3 analyzes the accuracy and robustness of the anomaly identification, anomaly classification and recovery critic for accidental anomalies. We study the recovery critic’s ability to re-enact reliably at different phases of the task. To this end, accidental anomalies were induced at specific graph phases as listed below:

- MV2PK: HC
- PK: HC, TC, OS, NO
- MV2PL: HC, OS

PL: HC

For the re-enactment recovery system, 60 recoveries were attempted (10 trials per object for 6 objects and induced by 5 trained users) on a per-node basis (4 total) under our two classification modalities: (i) perfect classification and (ii) imperfect classification.

6.4.4 Experiment 4. Testing Adaptation

Experiment 4 analyzes the robustness of the anomaly identification, classification, and adaptive recovery policy in the face of persistent anomalies. We analyze adaptation robustness by testing three scenarios with an increasing number of persistent anomalies (and thus adaptations). The three sub-experiments test robustness under the following conditions:

- 4a: one adaptation at a single phase (two examples).
- 4b: a 2nd adaptation introduced at a new phase.
- 4c: a 3rd adaptation introduced at a new phase.

For this experiment we run a total of 20 trials per persistent anomaly (4 objects with 5 trial runs per anomaly). A new anomaly class—Wall Collision—was discovered in these experiments and labeled (WC). We analyze whether adaptive policies work robustly independent of the number of adaptations that occur previously in the system and also whether or not the policies generalize across objects. Object locations and order are varied and randomized across trials. Sub-experimental details are given in three distinct sections below.

Experiment 4a: Adaptation at Distinct Single Nodes

In Experiment 4a, we analyze the robustness of the framework to properly identify, classify, and recover from persistent anomalies in single instances using adaptive recoveries. As described in Section 5.2, when the same anomaly occurs twice consecutively in the same node, the anomaly is considered persistent and an adaptive skill is learned from a user demonstration to recover and transition to the succeeding milestone in the task.

For this experiment, we tested two distinct persistent anomalies at independent phases of the task (node location is indicated by @# followed by anomaly type):

- 4a.1 : @2TC (pick).
- 4a.2 : @3WC (move-to-box).

Tool collisions (TC): occurred when two objects were placed by a human operator too close to each other. In such conditions, when the pick skill in node 2 is executed, one of the robot’s fingers collides with the neighboring object and prevents a proper pick as illustrated in Fig. 9. Re-enactments do not resolve the situation so help from a user is elicited to

Fig. 9 Persistent Pick Anomaly. On the left: the proximity of an adjacent object consistently precludes the proper gripping of a target object leading to a persistent tool collision. On the right: the execution of the learned adaptive skill which rotates the wrist and clears the fingers for the pick skill



overcome the persistent condition. The taught adaptive skill rotates the robot wrist about the approach axis and clears the fingers from the obstruction.

Wall Collisions (WC): in this (second) example, no tool collision occurs at node 2, however a persistent collision occurs at node 3 as the robot moves the picked object to the packaging box. The wall collision is a variant of a tool collision. Tool collisions were narrowly defined as collisions that occur on vertical downward motions. In this case, the collision occurs with a lateral motion and the contact can be either tool-wall (of the packaging box) or object-wall. The reason for such anomaly is that the original move-to-box skill was trained on an object of a given height and later a taller object was picked, the latter did not clear the wall with the original skill.⁹ Re-enactment does not resolve the anomaly; so an adaptive skill the clears the wall is taught as shown in Fig. 10.

Experiment 4b: Incremental Growth for Two Adaptations

In Experiment 4b, we analyze system robustness when two adaptive skills are learned incrementally for different phases of the task. It is important to ensure that the performance of the system is not compromised as more adaptations are introduced into the task graph. In this experiment, we integrate the adaptive recoveries learned in Exp. 3a and induce both persistent anomalies in the same experiment in an incremental fashion at different phases of the task:

4b: @2TC, @3WC.

In this way, the robot first responds by rotating its wrist to clear the persistent obstruction during the pick; and later upon collision with the wall, the robot responds by lifting its arm and clearing the box wall before placing the good in the package.

Experiment 4c: Incremental Growth for Three Adaptations

Finally in Experiment 4c, we analyze system robustness

⁹In. Section 7 we discuss possibilities for motion adaptation based on shape properties.

when we integrate the third adaptation. The next persistent anomaly occurs in node 4 as the robot places an object in the packaging box. The last anomaly results when, upon executing the placing skill, an object already in the box obstructs the placement of our currently held object. So the final sequence of anomalies at varying phase locations is:

4c: @2TC, @3WC, @4TC.

The Visualization module is in charge of allotting unique placement goals for all objects in a box, such that they all have a unique space within the package. However, it is possible that upon placement of an object, the latter falls and shifts to a different location in the box causing a tool collision. The adaptive skill teaches a simple displacement motion whose goal is parameterized by the visualization module to a clear location as shown in Fig. 11.

6.4.5 Experiment 5. Testing Re-enactment and Adaptation

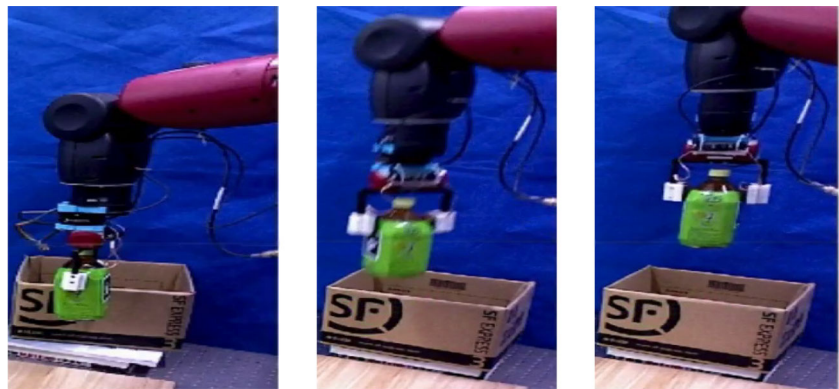
Experiment 5, analyzes the robustness of the system when re-enactment and adaptations are both integrated and present in the system. It is important to verify that re-enactment policies are not detrimental to adaptive policies and vice-versa. For this experiment, we integrate the accidental and persistent anomalies of experiments 3 & 4, and similarly use the re-enactments and adaptations already learned. Anomaly identification and classification metrics are presented as before under both classification modalities. The sequence of anomalies and recovery policies present in the system are delineated in Table 2, where we refer to re-enactments as “RE” and adaptations as “AD”.

For this experiment, 2 objects were selected at random and 10 test trials were conducted for each object. A total of 20 trials were run for each modality.

6.4.6 Experiment 6. Recovering from Anomalies that happen during Recovery

The final experiment analyzes the robustness of the system in identifying and recovering from anomalies (accidental and persistent) that occur during an ongoing recovery skill.

Fig. 10 Persistent Wall Collision. From left to right: Trained adaptive skill overcomes a wall collision via a clearing mechanism that lifts the object above the edge of the wall



It is imperative that the system performs reliably even during recovery actions. In this experiment, we test two situations:

- i. a persistent anomaly induced during an adaptation.
- ii. an accidental anomaly induced during an adaptation.

These two conditions will be referred to as “Adaptation over Adaptation” (AOA) and “Re-enactment over Adaptation” (ROA) respectively. Experiments are run under our two aforementioned classification modalities. Each experiment is executed for one object chosen at random and repeated 10 times. Details are shown in Table 3.

For (i) we use the same persistent anomaly and adaptation of Exp. 4a.1. Namely, during pick, one finger collides with the placement of an adjacent object. The original adaptation rotates the robot wrist about the approach axis by $\pi/2 rad$ (see Fig. 9b). In this experiment, we consider the placement of an additional object at the position where the already adapted grip fingers would descend. This in turn, would cause a new persistent tool collision. In this scenario, a new adaptation is needed. The human demonstrator decides to teach a sliding approach, whose direction of motion is parallel to the tangent of the table plane, until the fingers are centered on the object, at which point a pick behavior ensues. The adaptation is

illustrated in Fig. 12 and can also be seen in the video Supplement 1. For (ii) we combine the wall collision adaptation of Exp. 4a.2 with the phenomena experienced in Exp. 5 where an HC during move-to-place causes a subsequent OS that the system recognizes and one that is resolved via a pick re-enactment. In this case, we induce a human collision that results in a subsequent slip whilst the system is resolving a wall collision through a lifting adaptation.

6.4.7 Experiment 7. Anomaly Identification and Classification Re-activity

Exp. 7 analyzes the sensitivity of the anomaly identification to signals of different magnitudes and the sensitivity of the anomaly classifier to varying durations of the time window used to capture multi-modal signal observations before and after the anomaly identification flag has been issued. We wish to learn the limits in the reactivity of our algorithms. That is, how small can a signal be to flag anomalies or how quickly can we classify anomalies without sacrificing accuracy.

It is important to test the sensitivity of the anomaly identification system to signals of different magnitudes according to the setup we originally presented in Exp. 1.

Fig. 11 In the last phase, the robot attempts to place an object in the box but finds an existing object at the target location. A re-enactment does not solve the anomaly, so an adaptation is taught and a new goal provided by the visualization module



Table 2 Sequence of induced accidental and persistent anomalies into the system along with triggered re-enactment (RE) and adaptive (AD) policies during the Kitting experiment

Node	Anomaly Type	Recovery Type
2	TC	RE
3	HC	RE
3	WC	AD
4	TC	AD

Any identification system could be made arbitrarily accurate by overfitting. Here, we will measure the true positive rate of the identifier as well as the detection delay caused by changes in the anomaly signals. We follow the setup of [16] closely but with some variations. We first found the anomalous signal that had the largest amplitude in each of the phases of the task across all anomalies. This signal would then be inserted into a nominal signal (across all features), at a random time, with an amplitude ranging from 0-150% and then test for anomaly identification accuracy and detection time delays.

6.5 Results

In this section, we present detailed results and analysis for all experiments. In this section, we often times compare performance across nodes or specific settings. While comparing across experiments may be challenging since each node or setting has different complexity, we think that since the tasks were carefully designed to iteratively increase its complexity, we can glean meaningful insights across them.

6.5.1 Experiment 1. Anomaly Identification Results

Here we present a summary of the anomaly identification results for Exp. 3-6. Figure 13 charts the summary across nodes 1-4 and adaptive nodes that are generated when anomalies occur during recoveries as seen in Exp. 6. In Exp. 6, we analyzed two scenarios: Adaptations over Adaptations (AOA) and Re-enactments over Adaptations (ROA) which

Table 3 Conditions under which anomalies are induced during an adaptation

Events	Situation
@2TC-AD1, @2TC-AD2	AOA
@3WC-AD1, @HC→OS-RE	ROA

Node location for anomaly occurrence denoted with @; followed by anomaly type, and recovery policy indicated after (-). Also, (→) indicates a subsequently caused anomaly. For AOA: AD1 and AD2 describe 1st and 2nd adaptations. For ROA: RE refers to re-enactment

are discussed in detail there. All results and their analysis can be found in our Supplement 3.

Our anomaly identification accuracy for all experimental (766) trials was of 93.09% (see Supplement 3 for details). The precision was 94.09% and the recall 97.98%. These results show very strong accuracy and performance which is critical to avoid the aforementioned downstream errors.

In terms of performance across nodes, the experiments revealed very similar performance throughout the task with an average accuracy of 93.34%. This implies that anomaly identification performance did not improve or decline as the manipulation graph traversed the nodes—rendering the identification consistent and reliable.

The system also showed perfect accuracy and robustness for occasions in which persistent anomalies occurred during recoveries (AOA-Exp. 6). For times where accidental anomalies occurred during recoveries (ROA) the accuracy and precision was strong at 90% with no false-negatives.

We also compare our identification performance with a number of baselines consisting of probabilistic, traditional machine learning methods and deep networks. Baseline performance is compared across skills and over the whole task. For the probabilistic methods, we test HMM models with Gaussian ‘G’ and Autoregressive ‘AR’) observation models along with two inference algorithms (Expectation-Maximization ‘EM’, Variational Bayes ‘VB’). For machine learning, we use an Isolation Forest (iForest) algorithm and a Local Outlier Factor (LOF) algorithm. For deep networks we used a multi-layer perceptron with a variational auto-encoder (MLP-VAE) [45] and an Long short-term Memory (LSTM) network with a VAE [46, 47]. Parameter and hyper parameter settings can be found in Appendix E.

The results are seen in Table 4. The table shows how an HMM complexity of 10, which more closely resembles that of the sHDP prior, nearly achieves as good a result as our algorithm. Autoregressive observations help over Gaussian ones in every case. As for inference, Variational Bayes does better for sHDP-Gaussian observations, but slightly worse (0.921 v.s. 0.923) with an sHDP-AR observations. For more insights into the general performance of the inference algorithm, please refer to Fig. 17 in Exp. 2. Our algorithm achieves better performance compared to traditional machine learning algorithms as well as state-of-the-art deep networks like that of [46].

6.5.2 Experiment 2: Anomaly Classification Results

Anomaly classification accuracy across nodes (including the AOA and ROA nodes introduced in Exp. 1) is presented in Fig. 14. Note that we only consider trials in which the anomaly identification is correct such that the classification accuracy can be measured independently.

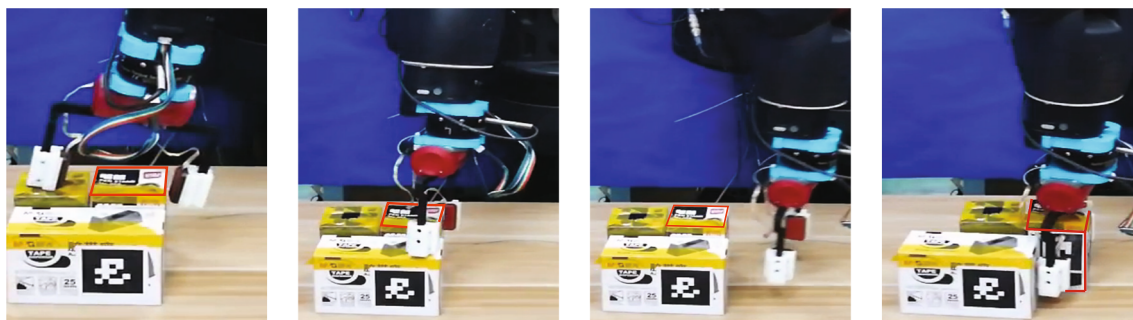


Fig. 12 Three objects in near proximity. Two smaller boxes towards the robot and a wider box away from the robot. Robot picks object delineated in red. A persistent anomaly occurs in Fig. 12a when a

finger collides with an adjacent box. An adaptation (Exp. 4a.1) is enacted but a new persistent anomaly (Fig. 12b) occurs with the wider box. Figure 12c & d show a newly taught adaptation

The same combination of features used for Anomaly Identification resulted in a sub-optimal performance but were maintained to prioritize the identification. Further tests could have been conducted to find the most optimal global combination for the system. The accuracy across the same distribution of features introduced in Exp. 1 is shown in Fig. 15. A confusion matrix was also computed for classification for all experiments and shown as a figure in Fig. 16. Furthermore, we used the F1-score metric to compare the performance variational inference algorithms across allocation and observation models. The models used for this comparison are listed below:

- Variational Inference Models: Memoized Variational Inference with Scalable Adaptation (MemoVB) and Variational Coordinate Ascent (VB).
- Allocation Models: HMM and sHDP-HMM

- Observation Models: Gaussian (Gauss) and the Vector-Autoregressive (VAR)

As for the variational inference algorithms, we compare the algorithm used in this paper; namely, memorized variational inference with scalable adaptation with variational coordinate ascent under different allocation and observation models. Stochastic variational inference was contemplated but not used as the algorithm did not converge after 1000 iterations. Gibbs sampling was also not used as it was not available as part of online BNPY [41]. The comparisons are also conducted as a function of the number of total training trials. The same number of total training trials was used as mentioned at the beginning of this experiment. Figure 17 shows the comparative performance of the inference methods.

Fig. 13 Summary of accuracy, precision, and recall metrics for anomaly identification across all experiments on a per-node basis, including recovery over recovery runs in Exp. 6, and a total summary of performance

ANOMALY IDENTIFICATION PERFORMANCE ACROSS NODES FOR ALL EXPERIMENTS (Exp 1)

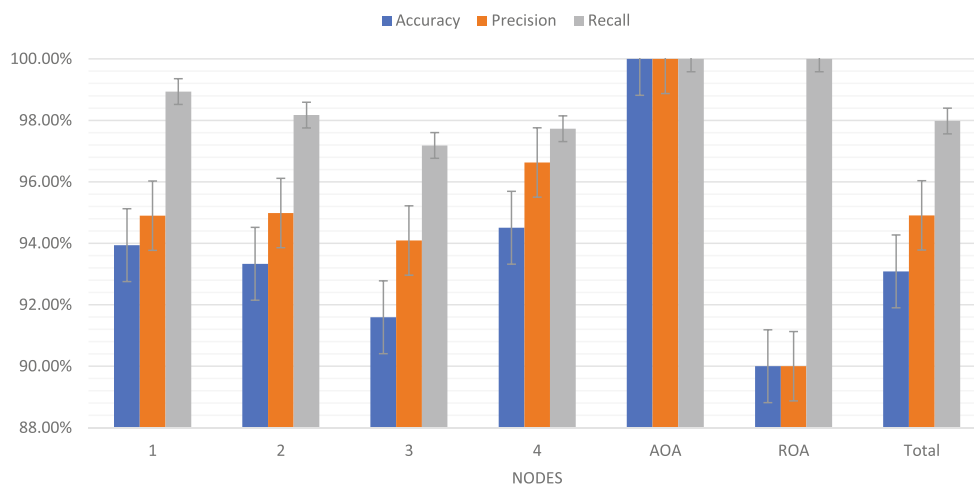


Table 4 Baseline comparisons for anomaly identification for each skill and overall task

Methods	k	#3	#4	#5	#7	#8	#9	Av
HMM-G-EM	3	0.96	0.89	0.75	0.78	1.00	0.78	0.86
HMM-G-VB		0.97	1.00	0.77	0.89	0.77	0.78	0.86
HMM-AR-EM		0.99	1.00	0.86	0.63	0.85	0.78	0.85
HMM-AR-VB		0.97	1.00	0.81	0.76	0.92	0.89	0.89
HMM-G-EM	5	0.99	1.00	0.78	0.65	0.92	1.00	0.89
HMM-G-VB		0.97	1.00	0.76	0.70	0.77	0.89	0.85
HMM-AR-EM		0.97	0.99	0.87	0.64	1.00	0.89	0.89
HMM-AR-VB		0.97	0.99	0.78	0.87	0.85	0.89	0.89
HMM-G-EM	10	0.97	1.00	0.78	0.80	0.92	0.89	0.89
HMM-G-VB		0.97	0.99	0.77	0.69	0.92	0.89	0.87
HMM-AR-EM		0.97	0.99	0.87	0.63	0.92	1.00	0.90
HMM-AR-VB		0.97	1.00	0.77	0.74	1.00	1.00	0.91
iForest		0.75	0.64	0.69	0.57	0.56	0.63	0.64
LOF		0.77	0.56	0.54	0.56	0.63	0.63	0.61
MLP-VAE		0.76	0.80	0.86	0.81	0.85	0.65	0.79
LSTM-VAE		0.83	0.87	0.77	0.88	0.72	0.97	0.84
HDP-G-VB	10	0.97	1.00	0.75	0.66	0.92	1.00	0.88
HDP-G-moVB		0.97	0.95	0.76	0.42	1.00	1.00	0.85
HDP-AR-VB		0.97	1.00	0.77	0.82	1.00	1.00	0.92
HDP-AR-moVB		0.99	1.00	0.79	0.76	1.00	1.00	0.92

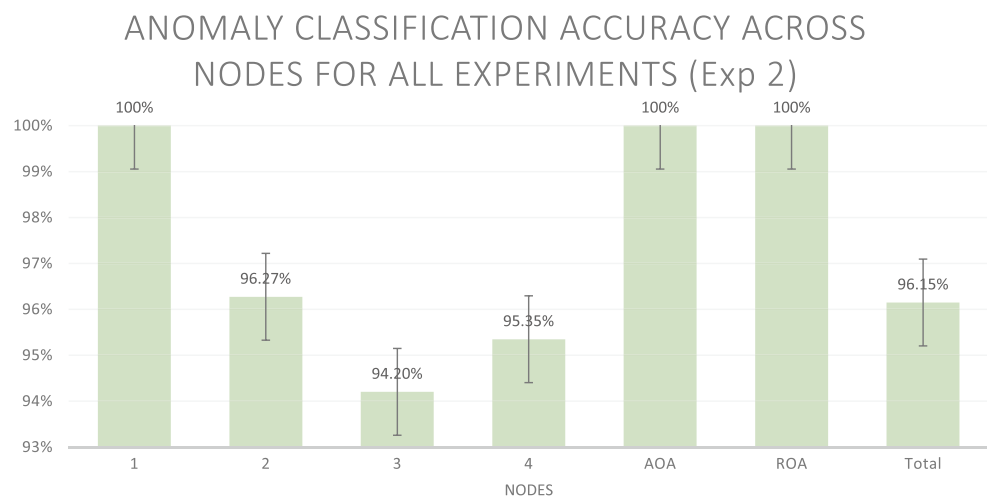
Red indicates ours

Best results appear in bold

Our anomaly classification accuracy for the totality of all experimental (719 trials) data was of 96.15% (Supplement 3). Interestingly, the accuracy of our anomaly classifier was overall more accurate than our anomaly identification routine. Extensive experimentation has been carried out. General trends are reported here, whilst specific experimental details are presented within each experimental section.

For the non-adaptive nodes, node 1 had perfect classification accuracy. Nodes 2-4 ranged from 94.20% to 96.27%. This indicates very similar performance over task-time and that the classifier was robust in detecting a varying range of challenges (see each experiment for specific details). The performance during ongoing recovery actions was of 100%. Although the number of trials for this section was 19, the

Fig. 14 Anomaly classification accuracy for all experiments across nodes 1-4 as well as adaptive nodes AOA and ROA created for anomalies under executing recoveries



Feat. Comb	#hc	#no	#os	#tc	#wc	#Avg.
1	0.000	0.538	0.537	0.444	0.211	0.247
2	0.895	0.609	0.915	0.588	1.000	0.765
3	0.778	0.933	0.679	0.625	0.722	0.721
4	0.889	0.714	0.759	0.706	0.947	0.766
5	0.292	0.824	0.712	0.368	0.167	0.357
6	0.481	1.000	0.880	0.412	0.818	0.642
7	0.783	1.000	0.815	0.611	0.944	0.714
8	0.739	0.765	0.688	0.722	0.333	0.573
9	0.556	0.786	0.657	0.700	1.000	0.654
10	0.684	0.429	0.745	0.722	0.708	0.553
11	0.500	0.778	0.800	0.176	0.895	0.588
12	0.167	0.933	0.587	0.765	0.636	0.548
13	0.833	1.000	0.846	0.941	0.947	0.815
14	0.611	0.786	0.729	0.941	1.000	0.756
15	0.670	1.000	1.000	1.000	1.000	0.859

Fig. 15 Anomaly classification accuracy for a distribution of features. Low, medium, and high probabilities are show in red, blue, and yellow respectively

data suggests strong classification performance even as the robot is adapting to anomalies.

In terms of the confusion matrix in Fig. 16, accuracy ranged from 88.7% to 98.4% for NO and HC respectively. The 2nd poorest classification was that of WC. WC were more challenging as the collision sometimes occurred against the gripper but in other occasions against the held object. OS came next with 93.1%, OS classification was challenged primarily by the tactile sensor noise experienced and explained later in Exp. 3.

With regards to variational inference performance, Fig. 17 shows how the sHDP-VAR-HMM with Memoized Variational Inference with Scalable Adaptation generally outperformed the rest of the combinations except for a couple of instances. In fact, in around 88.3% of the fraction of training trials our algorithm outperformed all others. The exceptions occurred roughly for the fraction 0.3-0.33 of the total training trials, where the sHDP-HMM-Gauss-MemoVB initially outperformed our algorithm 0.787 to 0.731. Similarly, for the fraction 0.87-0.90 of the total training trials, the sHDP-HMM-Gauss-VB outperformed our algorithm by 0.9%. Note that results will vary slightly across experimental runs as trial data is selected randomly and the probabilistic framework is unable to fix the random seed value across runs.

	HC	TC	OS	NO	WC
HC	98.4%	0.8%	0.8%	0.0%	0.0%
TC	0.0%	99.5%	0.5%	0.0%	0.0%
OS	6.3%	0.7%	93.1%	0.0%	0.0%
NO	0.0%	0.0%	11.3%	88.7%	0.0%
WC	3.1%	4.6%	0.0%	0.0%	92.3%

Fig. 16 Confusion matrix computed in Exp. 2 for all occurring anomaly classes in the Kitting experiment across all experiments

6.5.3 Experiment 3: Re-enactment Results

The results for anomaly identification and anomaly classification for Exp. 3 are shown in Fig. 18a & b. A confusion matrix for classification accuracy is shown in Fig. 19.

The result of the re-enactment policy for modality (i) is shown in Table 5 and for modality (ii) in Table 6.

For anomaly identification, a total of 574 trials were used for testing (103, 265, 144, and 62 for nodes 1 to 4). An average accuracy of 91.16%; a maximum of 93.94% and a minimum of 87.14% in nodes 1 and 4 respectively. For precision we had an average of 93.42%; a maximum of 94.90% and a minimum of 91.04% in nodes 1 and 4 respectively. For recall we had an average of 97.37% with a maximum of 98.94% and minimum of 95.31% in nodes 1 and 4 respectively. The reason node 4 may experience lower robustness might be due to the fact that more variations exist over time (e.g. poses may vary in ways that modify the previously experienced dynamics during training).

For anomaly classification, a total of 516 trials were used for testing (93, 242, 122, and 59 for nodes 1 to 4)with an average accuracy of 96.87%. Nodes 1 and 2 were classified perfectly, followed by 4, and struggled the most with node 3 at an accuracy of 90.77%. The confusion matrix for anomaly classes shows perfect or near perfect classification for TC and HC respectively and struggled more with OS and NO as shown in Fig. 19. OS detection suffered primarily form noise in our tactile sensor. We believe a large portion of the noise came from false contacts in the electronics in the tactile sensor. Whilst we attempted to rigidly fix the sensor’s electronics, there was still wiggle during anomalous events. With regards to NOs, we were surprised with the lower classification rate. We believe that the tactile sensor’s noise was also the culprit. We wanted to use the infra-red sensor on the robot’s wrist as an additional observation source, however, the force-torque sensor set-up blocked the IR signal and prevented its use.

With regards to re-enactment recovery, we present success rates for both classification modalities. Under perfect classification, we re-enacted and completed the task successfully on average 98.75% across all nodes (see Table 5). Some failures occurred in MV2PL as an OS occurred. After the OS, the object reached a location outside the field of view of the camera and prevented the system from computing the object pose.¹⁰ Under imperfect classification,

¹⁰We should note that there were 11 other trials where system failures occurred (these were not marked as recovery failures). There were two main causes for the system failures: (a) challenging pick poses resulted

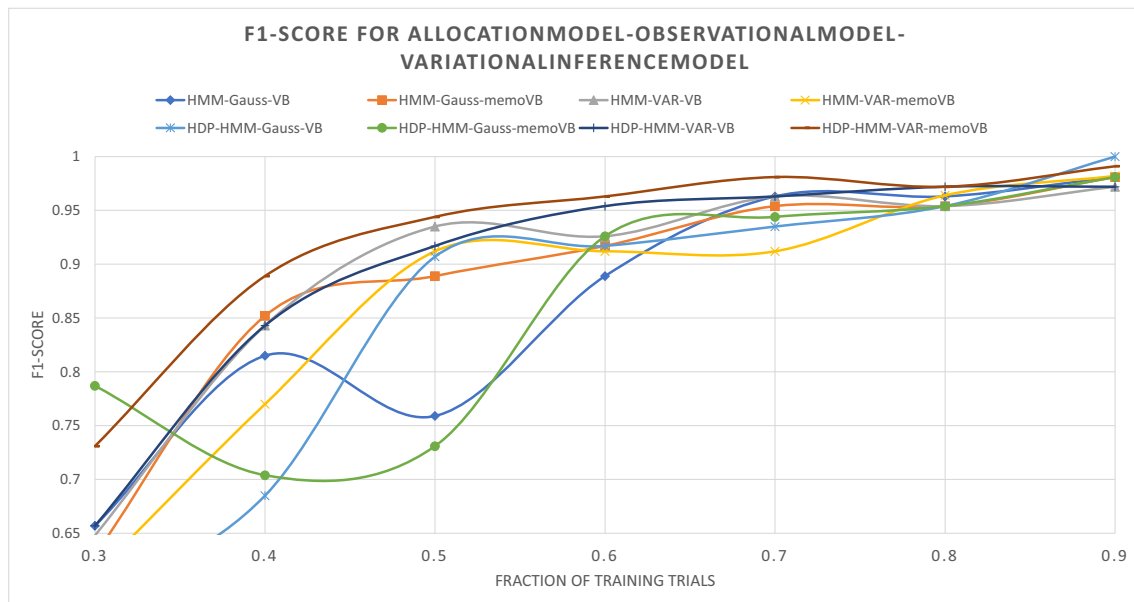


Fig. 17 F1-score metric performance comparison for variational inference algorithms across allocation and observation models

we expected a lower performance, and obtained an average recovery completion of 92.81% across all nodes (see Table 6). The highest rates were obtained in node 1 and 4 under HC anomalies with 95% recovery, TC anomalies in node 2 with 98.3% recovery, and OS anomalies in node 3 at 95% recovery. The picking skill was the most problematic to resolve in the presence of HCs and NOs.

With regards to overall system trends we observe: very competitive anomaly detection at an average of 91.16% and very high anomaly classification (one of our contributions) at 96.87% as shown in Fig. 18. For re-enactment under independent systems we see that re-enactments can resolve almost all accidental anomalies at 98.75%.

One last but very interesting development was evident when we computed the performance of the *entire system* under the two classification modalities as seen in Fig. 20. Note, interestingly, that for one out of the four nodes—node 3—the overall success rate of the combined system was higher than that of the independent system. This implies that system completed the task successfully more times under imperfect classification than with perfect classification. The specific reason for this phenomena is that soon after a misclassification takes place; the introspection system detects that the robot is still in an anomalous state and triggers a new anomaly flag and issues a new round of classification. This time the correct policy is issued and resolves the anomalous situation. One example is when an OS was misclassified as an HC. The HC triggers a re-enactment, but the robot is not grasping the object. At a later time step,

the introspection system flags another anomaly and classifies it as an NO. This time a pick re-enactment is issued and enables the robot to successfully complete the task.

6.5.4 Experiment 4: Adaptation Results

In Experiments 4a,b,c, for anomaly identification, a total of 124 trials were used for testing.¹¹ For anomaly identification, we had an average accuracy of 97.04%, an average precision of 97.02% and an average recall of 99.42% across the three sub-experiments. Very strong performance was achieved all around and charted in Fig. 21a.

For anomaly classification, a total of 121 trials were used for testing¹² with an average accuracy of 94.09%. Experiment 4a.2 had the worst performance at 85.0%, followed by Experiment 4b at 94.59%, and perfect classification in Experiment 4c.

A confusion matrix was computed for classification accuracy and shown in Fig. 22. TC and WC are the core classes, whilst HC appears as a result of misclassification.

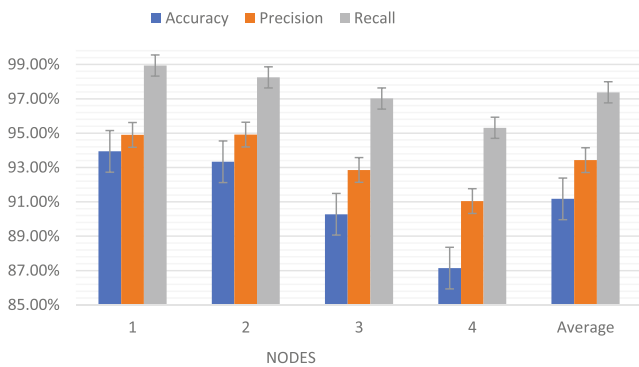
Across all sub-experiments we were able to identify TCs in Exp. 4a and 4c with 100% accuracy. Wall collisions were less accurate at 89.80%. Wall collisions were harder to classify given that those collisions occurred under two different scenarios: at times the gripper collided with the box and at other times the held object made the collision. Hence, the multi-modal signals contained variations that degraded the classification performance.

in tactile sensor cables constraining the gripper and (b) an electricity overload in the system that rendered parts of the robot to a halt.

¹¹20, 21, 38, and 45 for experiments a.1, a.2, b, and c.

¹²20, 20, 37, and 44 for experiments a.1, a.2, b, and c.

Anomaly Identification for Accidental Anomalies (Exp 3)



Anomaly Classification for Accidental Anomalies (Exp 3)

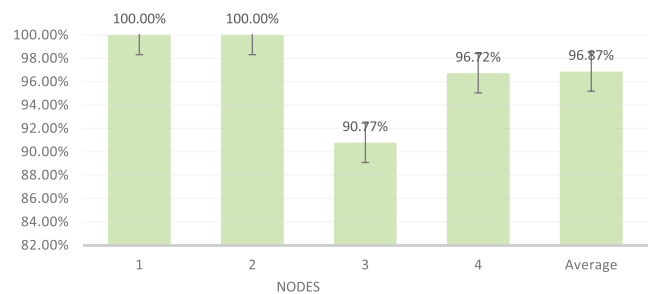


Fig. 18 Accuracy, precision, and recall metrics for the anomaly identification and accuracy metrics for the anomaly classification system on a per-node basis for accidental anomalies (left and right respectively)

With respect to adaptive recoveries, Fig. 23 presents success rates under our two classification modalities. As expected, the success rates under perfect classification generally were higher than those with imperfect classification with an average across sub-experiments of 85.0% and 77.5% respectively. The exception was Experiment 4a.1, where the imperfect classification modality achieved 95.0% success rates v.s. 90.0% for modality (i). The failures under modality (i) were due to manipulation system errors. In one trial, during the move-to-box node, the object’s collision with the packaging box moved the latter and the place action failed. Our system is limited by not actively tracking objects of interest and rationalizing relationships between them (see Section 7 for more comments on this).

The results also reveal that one object-set of trials in Experiment 4c had difficulties (see Appendix E for comments). If each of those two trial-sets were not considered, the average success rate would be to 90.83% and 82.50% for perfect and imperfect classification modalities respectively.

With respect to overall system performance, we again compare the performance between modalities. We achieved an average success rate of 78.02% and 75.36% for both modalities respectively. Figure 24 charts the results over sub-experiments and modalities.

	HC	TC	OS	NO
HC	99.2%	0.8%	0.0%	0.0%
TC	0.0%	100.0%	0.0%	0.0%
OS	4.8%	0.8%	94.4%	0.0%
NO	0.0%	0.0%	11.3%	88.7%

Fig. 19 An anomaly classification confusion matrix for accidental anomalies in Exp. 3

As with Exp. 3, we again see the interesting phenomena that for Experiment 4a.1, modality (ii) achieved higher success rates than modality (i). It supports the premise that even when there are misclassifications in the system, the task can be completed as the system some time later correctly detects, classifies, and recovers from existing anomalies.

6.5.5 Experiment 5: Test Re-enactment and Adaptation Results

Anomaly identification results across nodes can be seen in Fig. 25a while anomaly classification accuracy can be seen in Fig. 25b. The anomaly confusion matrix is shown as a figure in Fig. 26. Corresponding success rates for modalities (i) and (ii) are summarized in Table 7. Notation in Table 7 has been abbreviated as follows: for a TC that occurs at node 2 followed by a re-enactment, the notation we use is: @2TC-RE, hence @ indicates the node phase, followed by the two digit anomaly, followed by a dash to indicate the type of recovery. Tables 8 and 3 follow the same notation.

We now summarize the results for Experiment 5. For anomaly identification a total of 72 trials were tested. We had an average accuracy and recall of 97.9% and a perfect precision. For nodes 2 and 3 anomaly identification was

Table 5 Recovery Success Rate w/ Perfect Classif—modality (i) across nodes & anomaly classes

Node	HC	TC	OS	NO	Average
1	100%				100%
2	100%	100%	100%	100%	100%
3	100%		90.0%		95.0%
4	100%				100%
Total					98.75%

Table 6 Recovery Success Rate w/ Imperfect Classif—modality (ii) across nodes & anomaly classes

Node	HC	TC	OS	NO	Average
1	95.00%				95.0%
2	85.00%	98.3%	88.3%	80.00%	87.9%
3	91.7%		95.0%		93.3%
4	95.0%				95.0%
Total					92.8%

done perfectly for the three metrics. It was node 4 that was more challenging with an accuracy and recall of 93.8% and perfect precision. For anomaly classification, 71 trials were tested with an average accuracy of 92.8%. As with anomaly identification, it was also node 4 that was the most challenging to classify followed by node 3 with an accuracy of 86.7% and 91.7% respectively. Note that by the time the robot reaches node 4 it has undergone 3 different anomalies and is undergoing one more and the system has also experienced two re-enactments and an adaptation. As discussed earlier, a high degree of variability in the sensory-motor signals (compared to training) begins to enter the system as more recoveries take place and change gripping poses, dynamics and inertia, and the interaction with the objects.

With regards to success rate, under classification modality (i) the success rate was 90.0% and under modality (ii) the rate was 80%. Fatalities occurred during the wall collision where the collision caused an object slip that displaced the object beyond the camera's field of view impeding any further attempts to re-pick. Under imperfect classification, we experienced a misclassification of HC as OS. The robot attempted to re-enact a pick. However, the object's pose was too high and no IK solutions existed. On another occasion a WC got misclassified as TC repeatedly, we aborted after 3 attempts. Specific experimental outcomes can be found as

comments for this experiment in Supplement 3, under the "Exp 5" tab in Excel.

The Wall collisions experienced in this experiment, afforded a new phenomenon. Namely, how the generation of one anomaly leads to the trigger of a subsequent anomaly. In Table 2, note that an HC is induced in node 3. This same HC can trigger an OS in the task. For this reason, we further studied the system's ability to recover from a subsequent OS anomaly. As before, 10 trials were tested for the same 2 objects under both classification modalities with results shown in Table 8.

Under perfect classification 90.0% success rates were also achieved. The fatality occurred when the wall collision displaced the packaging box in a way that precluded further placing of objects in the box. For imperfect classification 70.0% success rates were achieved. In this experiment, during node 3, when an OS occurred, the system misclassified as a HC and triggered a re-enactment of the same node. Later the system triggers an NO object flag; however, because we had not previously trained a re-enactment at node 3 (only for node 2) the system halted. Experimental details can be found as comments can also be found under Supplement 3.

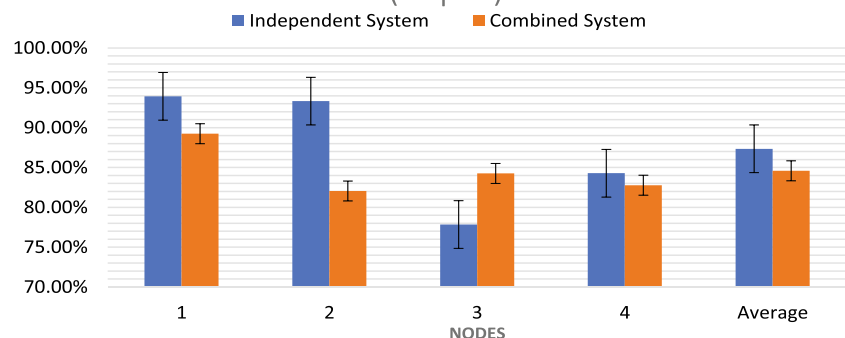
6.5.6 Experiment 6: Recovering from Anomalies that Happen during Recovery Results

For anomaly identification, a total of 20 trials were used for testing (10 and 10 for experiments AOA and ROA) and had an average accuracy of 100% and 90.0% for AOA and ROA respectively. Precision had the same performance and recall was perfect.

For anomaly classification, a total of 19 trials were used for testing (10 and 9 for experiments AOA and ROA) and had an average accuracy of 100% and 77.78% for AOA and ROA respectively. A confusion matrix was also computed and shown as a figure in Fig. 27. TC and WC were the

Fig. 20 Overall system success rate as a function of modality. Modality (i) considers identification (AD), classification (AC), and recovery (REC) as independent systems. Modality (ii) considers an independent AD system with a combined AC/REC system. MV2PL shows interesting phenomena: the 2nd modality performed better than the 1st as wrong classifications were corrected downstream and coupled with correct recovery policies

Overall System Success Rate for Re-enactments under different Modalities (Exp 3)



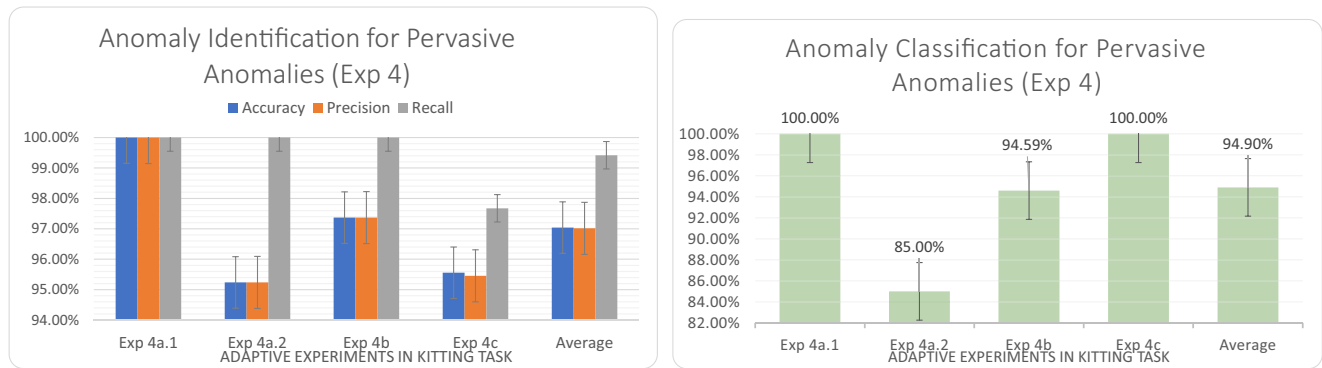


Fig. 21 Accuracy, precision and recall metrics for the anomaly identification system and accuracy metrics for the classification system on a per-(sub)experiment basis for persistent anomalies (left and right respectively)

target classes and the resulting HC statistics were due to miss-classification.

As for success rates, each of the two situations under both classification modalities are shown in Table 9. For Adaptations-over-Adaptations, the system success rates were 80.0% and 90.0% for both classification modalities respectively. For AOA with perfect and imperfect classification, a total of three failures occurred (details are found in Appendix E.

For Re-enactments-over-Adaptations, the system success rates were 100% and 70.0% for both classification modalities respectively. The latter was caused by 1 false-negative in anomaly identification, 1 false-positive in node 3, and the same system limitation previously mentioned for AOA also occurred once here. If we look at the combined contribution of both situations for a given modality we have 90.0% for perfect classification and 80.0% for modality 2.

6.5.7 Experiment 7: Anomaly Identification and Classification Reactivity Results

Figure 28 shows that our anomaly detector is generally insensitive to changes in the amplitude of the anomalous signal. Except for skills 4 and 9, skills show a TPR of above 80% for amplitudes one-fourth of the original size. Such performance is better than that of [16]. The lower performance for the other two skills is due to a lower

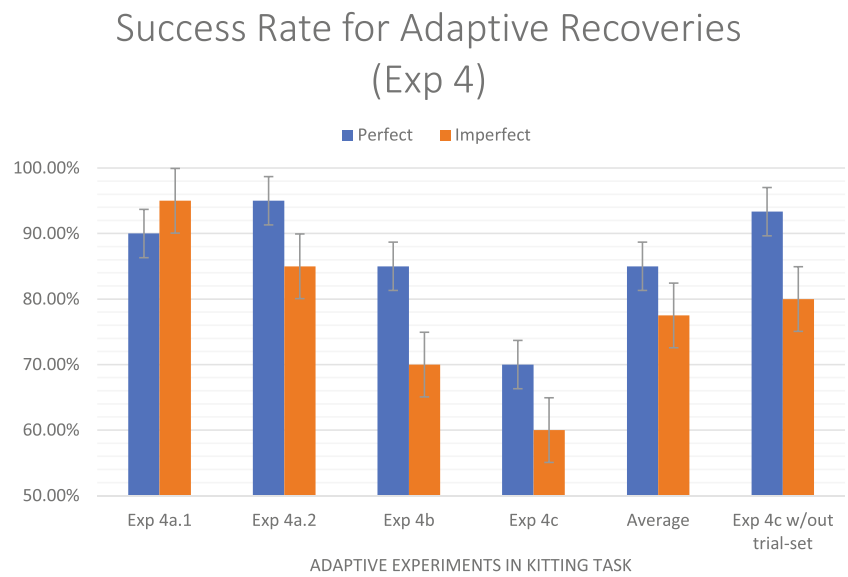
	HC	TC	OS	NO	WC
HC	0%	0%	0%	0%	0%
TC	0%	100%	0%	0%	0%
OS	0%	0%	0%	0%	0%
NO	0%	0%	0%	0%	0%
WC	4.1%	6.1%	0%	0%	89.8%

Fig. 22 An anomaly classification confusion matrix for persistent anomalies TC and WC in Exp 4

number of anomalies acquired during those phases of the task. For larger signals, the performance seems largely consistent. In terms of time delay, we also show a very consistent performance. Our average delay remains below 0.5 secs until the signal becomes one-fourth of the original size. This performance also surpasses that of [16]. With respect to the anomaly classification sensitivity to the signal time duration window, recall that we use a standard windows of ± 2 seconds to capture observations before and after an anomaly has been identified (Section 2.2). Figure 29 shows a contour map of anomaly classification accuracy as a function of pre and post anomaly identification time duration. The figure contains accuracy regions in groupings of 5 percentile points, where the lower left corner indicates the smallest range of time windows, whilst the top right corner indicates the longest range time windows. The anomaly classification data in this experiment was setup in the same way as in Exp. 2. The final anomaly classification accuracy is computed as the average of the true-positive confusion matrix rates. Finally, note that reactivity measurements for anomaly identification were originally presented in [17] and concluded that we could identify anomalies on average consuming 1.84% of the duration of skills.

According to Fig. 29, classification accuracy seems to be the highest (95% and above) in an approximate golden central radius, with another outer ring in gray holding the next percentile accuracy grouping (90-95%). For the smallest window combination, the lower left corner, the classification accuracy ranges in the (80-85%) grouping. Recall from Exp. 2 that our overall anomaly classification accuracy for the standard ± 2 second window was of 96.15%. The contour patterns seen in our experiment indicates that in general there tends to be quite similar performance in most of the studied regions. Only the region from 0.5-1.0 seconds seems to register a symmetrical drop in performance across both axis from the 90-95% range to

Fig. 23 Success rates for Adaptive recoveries under two modalities: (i) perfect classification and (ii) imperfect classification



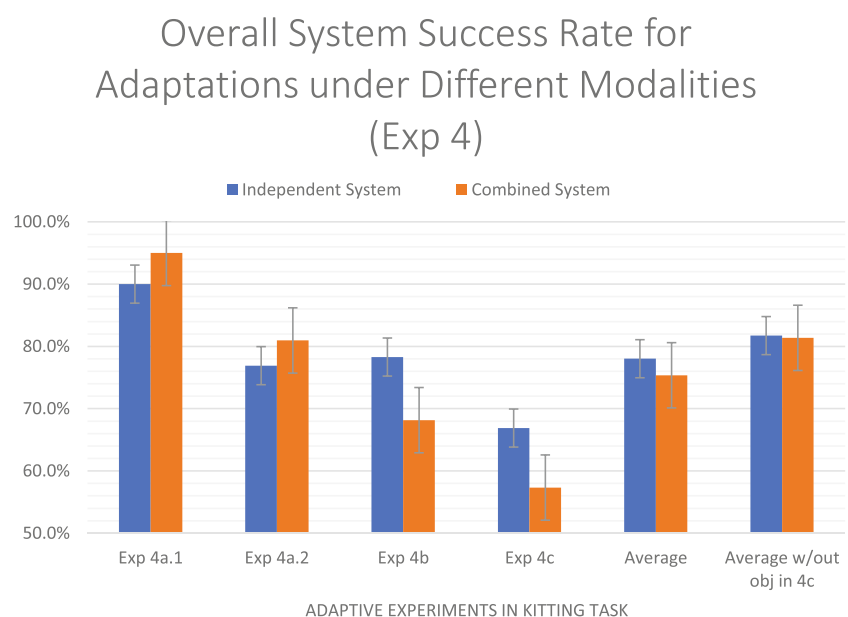
the 80-90% range. Such information indicates that the main structural signatures of anomalies require slightly more than one second, given our classification algorithm in this kind of task, to provide accuracies above 90%. Note that the Supplement 1 video uses the standard time window capture of ± 2 seconds.

6.6 Summary

In this last section we summarize and analyze the performance of the recovery policies. Figure 30 shows the success rate across experiments along with the final percentage as a total sum across all experiments. Comparing across experiments has limitations as each task has

a different level of complexity. When we consider classification modality (i), we can isolate the recovery critic performance. When considering counts across all experiments, our system was able to successfully recover 96.33% of the time. This result is not the experiment's average and reflects the more heavily weighted results of Exp. 3 where we had 98.75% success rate for re-enactments across 480 trials (across nodes, objects, and users). If we consider the average performance, we still obtain a very strong 92.02%. This result reflects what we have commented on already: a larger number of recoveries induces larger variability in observations making introspection more challenging (we recovered 85% of the time in Exp. 4). Nonetheless, we still recovered on nine out of ten

Fig. 24 Overall system success rate as a function of modality for adaptive recoveries. Modality 1 considers perfect classification and modality 2 considers imperfect classification. It is surprising that some experiments with imperfect classification outperformed those with perfect classification in success rate. Wrong classifications were corrected downstream and coupled with correct recovery policies



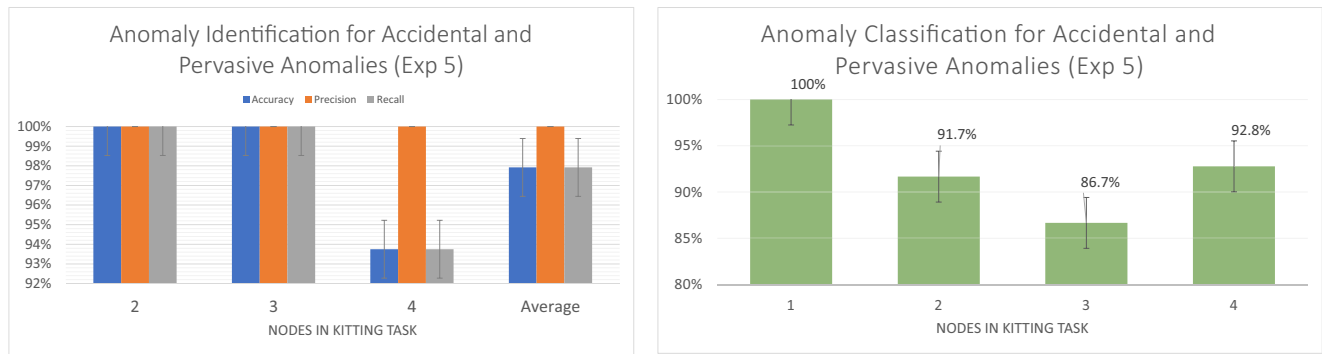


Fig. 25 Accuracy, precision and recall metrics for the anomaly identification system and accuracy metrics for the classification system on a per-(sub)experiment basis for merged accidental and persistent anomalies (left and right respectively)

times across users, objects, anomaly types, and nodes in the graph, hence showing very strong performance overall.

When we consider classification modality (ii), we are considering the entire system and the effects of not only the recovery critic, but also those of anomaly identification and anomaly classification. These results describe the effectiveness of a highly integrated introspection and recovery system (along with a manipulation and visualization aspects of the framework). When consider all counts across experiment we recovered 88.33% of the time and we consider the averaged result 82.38% of the time. Hence, the integration of the complete system, diminishes the performance of the recovery system, by slightly less than 10% points. Again, within comments we emphasized that the loss in performance was mainly experienced in Exp. 4 and 5 where a large number of anomalies were induced. This will often not happen in practice. Exp. 4a might be a more likely event, where 95% recovery was achieved under imperfect conditions in our work. Exp. 5 contained our worst performance with successful recoveries 75% of the time. This may not be a bad result after all. Recovering more than seven times out of 10 with unexpected scenarios, in our estimation, is not bad for current robotic performance in unstructured environments. Furthermore, in Section 7, we comment in detail specific directions in which we can significantly improve and expect better results. All experimental data is contained in Supplement

2, results analysis can be found in Supplements 3 and 4, and code in Supplement 5. We expect the community to use the current work and results as future baselines and improve performance further.

7 Discussion

Our comprehensive experimental results showed that our tightly-integrated, graph-based online motion-generation, introspection, and incremental recovery system worked accurately and robustly for a wide range of anomalous situations in an unstructured co-bot scenario where a human and a robot collaborated to complete kitting tasks. To the best of the author’s knowledge, this is the first study where the recovery ability of a robot is examined in the presence of anomalies in manipulation in unstructured environments. In our study, we demonstrated that we could not only identify anomalies reliably (overall accuracy of 93.09%) but also classify them in an online fashion (overall accuracy of 96.15%). And that given simple task-level recovery policies, we could also recover consistently and reliably most of the time. The tight integration achieved in this work enabled robots to continue functioning, more than 82% across all our anomaly scenarios, and 95% in more typical scenarios like Exp. 4a. Even when anomalies occurred during recoveries

	HC	TC	OS	NO	WC
HC	80.0%	0%	20%	0%	0%
TC	0%	97.1%	3%	0%	0%
OS	10.0%	0%	90.0%	0%	0%
NO	0%	0%	0%	0%	0%
WC	0%	0%	0%	0%	100%

Fig. 26 An anomaly classification confusion matrix for accidental and persistent anomalies HC, TC, OS, and WC in Exp. 5

Table 7 Success rate for combined Re-Enactments & Adaptations for 2 objects under different classifications

@2TC-RE @3HC-RE @3WC-AD @4TC-AD	Success Rate
Modality (i): Perfect Classification	90.0 (10.0)%
Modality (ii): Imperfect Classification	80.0 (10.0)%

Node location for anomaly occurrence denoted with @; followed by anomaly type. Recovery policy indicated after (-). System errors as a % of total failures is enclosed in parenthesis

Table 8 Success rate for combined Re-Enactments & Adaptations in the presence of a subsequently generated anomaly for 2 objects under different classification modalities

@2TC-RE @3HC-RE @3WC-AD→OS @4TC-AD	Success Rate
Modality (i): Perfect Classification	90.0 (10.0)%
Modality (ii): Imperfect Classification	70.0%

Generated anomaly denoted with(→). System errors enclosed in parenthesis as a % of failure contribution

themselves, we recovered with 80% of effectiveness. Hence, the combination of anomaly identification, with global classification and simple but contextual task-level policies reliably showed broad robustness in being able to recover at all stages of the task, across all anomaly conditions, across different users and objects thus extending the autonomy of the system in significant ways. While the system has a number of weaknesses we will soon address, this system with simple observation capabilities of the world may serve robotics systems where sensors are limited but desire more robustness in unstructured environments.

A couple of unexpected but welcome results are also discussed. First, the robustness results of the anomaly classification system and the recovery critic were somewhat unexpected. The sHDP-VAR-HMM model displayed a strong ability in generating good models that worked across different phases of the task and identified anomaly categories that contain important variations within. The limits of the model seemed to have shown up in Exp. 5 at node 3, when the most strenuous conditions were presented. Even there the classification system had an 86.7% accuracy. In our hand-engineered features, we attempted to abstract structure from the data instead of only keeping raw-observations. Such that, if signal patterns that were similar occurred at dissimilar temporal positions during the observation window, they would still possess similar representations. Structure was abstracted by integrating the norm of each of the modalities in our feature set.

The second unexpected emergent result occurred when we presented results for classification modality (ii) and saw that the combined (AD/AC/REC) system at times had better performance than under modality (i) where we had perfect classification (see Exp. 3, node 3, in

	HC	TC	OS	NO	WC
HC	0%	0%	0%	0%	0%
TC	0%	100%	0%	0%	0%
OS	22.2%	0%	77.8%	0%	0%
NO	0%	0%	0%	0%	0%
WC	0%	0%	0%	0%	0%

Fig. 27 An anomaly classification confusion matrix for persistent anomalies TC and OS in Exp 6

Table 9 Success rate for anomalies and adaptations that occur during the original execution of a recovery policy

Situation	Perfect	Imperfect
AOA	80.0 (20.0)%	90.0 (10.0)%
ROA	100%	70.0%
Total	90.0%	80.0 (10.0)%

One object and two classification modalities are used to report performance metrics. System errors enclosed in parenthesis as a percentage of failure contribution

Fig. 20 and Exp. 4a.2). There we learned that *many anomaly misclassifications did not result in unsuccessful task completions*. We learned in fact that the system could self-heal. Even when a misclassification was originally

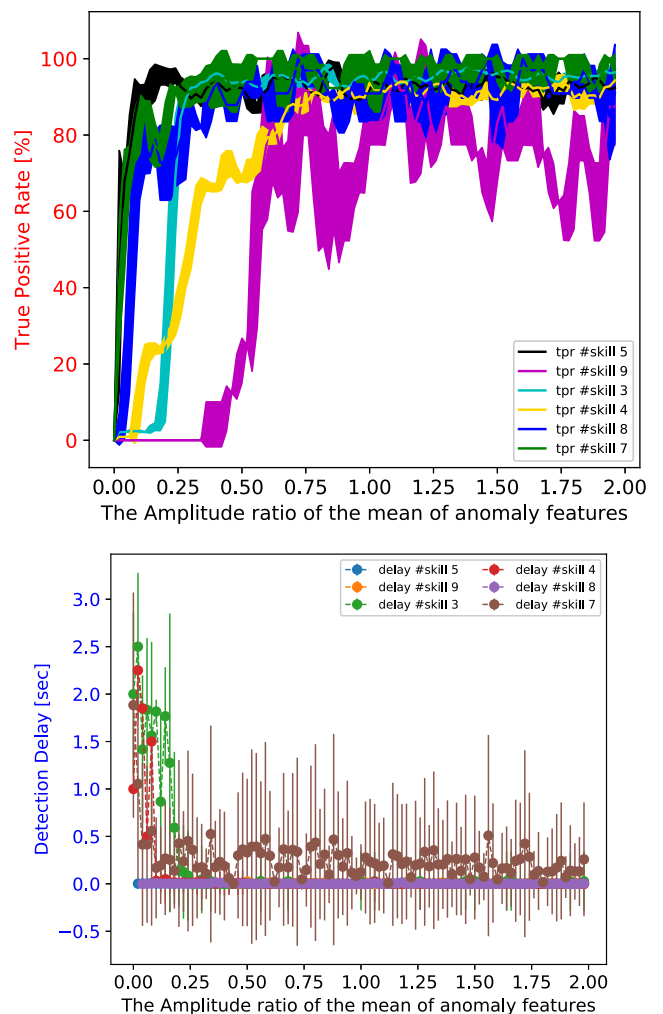
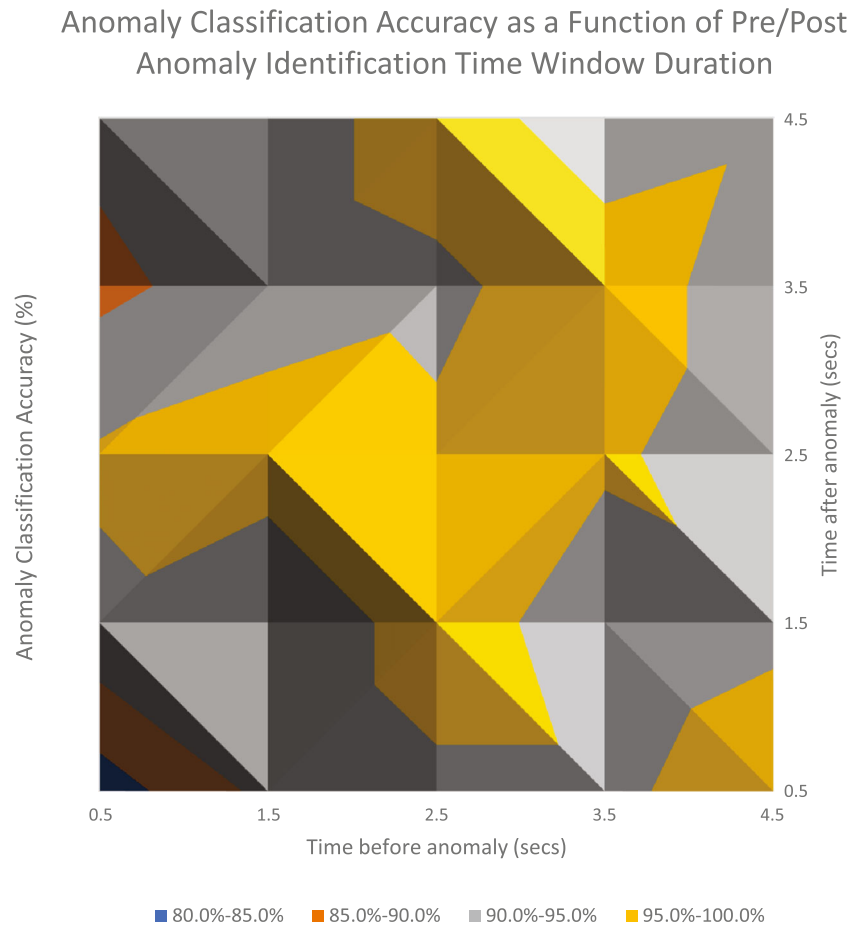


Fig. 28 Anomaly detector sensitivity to changes in simulated anomaly signal amplitudes. After identifying the maximum anomaly signal for each phase of the task, we randomly inserted it to a nominal signal with signal ranges from 1 to 150% and study the true-positive rates and the detection delays variations

Fig. 29 Contour plot of anomaly classification accuracy given pre/post anomaly time window duration. Bottom axis: captured time before anomaly flag. Right axis: captured time after anomaly flag. The plot presents accuracy regions in groupings of 5% points. The broad golden central area shows that we achieve high accuracy across large time window duration variations

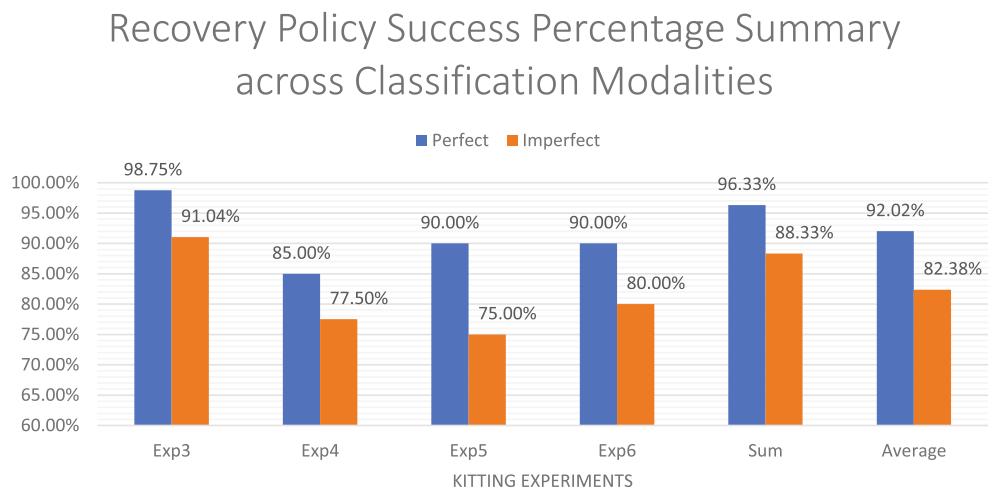


present and an inappropriate recovery policy enacted, the system self-corrected at a later time step by correctly understanding its anomalous state and later triggering the correct recovery policy.

We believe this work has broad applicability. Its graph based structure with internal modules for motion generation and introspection, and a supervisory recovery critic, allow

the system to leverage any class of motion generation algorithms including attractor-based, probabilistic, and deep end-to-end approaches (better introspection techniques can be leveraged as well). The bottom-line is that even as motion generation techniques become increasingly robust to disturbances [48–50]; failure is still a frequent occurrence when uncertainty in the environment surpasses the modeling

Fig. 30 Recovery policy success rate across experiments along with final rate across all experiments for both classification modalities



ability of the system. Thus, our framework can enhance the long-term autonomy and robustness of systems that use various motion-generation approaches. Furthermore, we believe this system would may work in manipulation scenarios with higher precision demands. In [25], we effectively classified anomalies for snap assemblies with complex dynamics and tight tolerances. However, as we will mention later, it is critical to automate the recovery policy learning portion in this work.

Additionally, the deep system integration presented in the paper allowed for a comprehensive study of the dynamics between an introspection system and an accompanying recovery-critic. We believe this is the first study of its kind, where an explicit and detailed study of the anomaly-recovery relationship is presented. We have open-sourced the code, dataset, and result analysis (see Supplements 5, 2, and 3/4 respectively) to promote and facilitate further examination of the topic. We hope others can build on our work and use the current results to further improve performance. There is still much improvement ahead and we attempt to discuss some of the main issues next.

7.1 Limitations, Comparisons, and Future Work

An important limitation in our work is the fact that the kitting experiment was not conducted under real warehouse conditions. Thus the applicability of the work in real-world applications is unclear and further testing in real-warehouse conditions is necessary. The kitting experiment provides a proof-of-concept and the authors would like to extend their work to actual scenarios through corporate partners.

With regards to re-enactment policies on a task-planning level, the multinomial distribution is admittedly simplistic. It is an indirect process of capturing decision policies. Furthermore, while we try to reduce re-teaching by having adaptation nodes inherit re-enactment policies from their parent node; there are times anomalies will occur for the first time in later nodes for which no policy exists. This requires user intervention to train the system as happened in Exp. 5 for imperfect classification where the system halted its performance because no re-enactment policy existed for the NO class in a particular node. We are interested in looking for automated policy learning solutions that evolve over time. The tight integration of object-centric motion planning may be able to improve or supplement re-enactment performance. This is part of our future work.

With regards to adaption policies, we do not yet model the spatial relations amongst the actors of interest; namely, the robot (end-effector), active objects (like objects to be gripped and the packaging box), and the world (support surfaces like tables and floor). These relationships provide

important context for decision making and are recently attracting more attention [51–54]. Without spatial relation understanding, the solutions learned in Exp. 6 will not extend to situations where the spatial relations are different from those in training. Not all experiments would fail without spatial relations context however. The HC, OS, and NO anomalies do not seem to explicitly depend on spatial context and may likely be resolved as-is in new situations. In effect, despite the lack of explicit spatial relationship modeling, our recovery policies often overcame external disturbances that might have otherwise terminated the task and endowed the system with longer operational horizons. By learning context relations, adaptations would do more than replay a learned behavior, they would in fact restore the complete and original state of the system before the anomalous condition. The larger overall challenge remains in learning how to integrate real-time reasoning and apply it to a learned skill, how to explicitly consider the spatial and functional relations between objects, the robot, and the world. It is possible that by theoretically grouping anomaly-recovery pairs into groups that do need functional-spatial reasoning and groups that do not [55, 56]. In [57, 58], for example FOON graphs and object affordances are introduced and might be particularly useful to resolve spatial and reasoning problems. Resolving this issue will be a consideration for future work. Notwithstanding, the work as-is with its limitations, might be useful in extending the autonomy of robots with limited sensor and/or computational capabilities.

With regards to anomaly identification, the work of Park et al. [15, 16, 18, 46] is the most closely related to our work.

For Park et al., there are a couple of comparison points to be made. The first point relates to the way anomaly data is compartmentalized. Their system applied HMMs to identify anomalies for ensembles of either: a specific robot skill with a specific object, or a specific robot skill with a specific person. Such specificity makes it easier to identify anomalies but it also increases the number of classes to be trained. Evidently, models that can accurately discriminate across broader datasets (such as being trained with a multiplicity of objects or users) is desirable. In our work, our anomaly identification (and classification) was trained to identify anomalies across different task nodes, different objects, and different users (where relevant). Thus, a broader training domain was considered in our work. Our anomaly detector achieved better performance than the LSTM-VAE of [46] and was more insensitive (more accurate) and faster compared to the HMM-D of [16] as shown in Exp. 1 and 7.

With regards to anomaly classification our system seems to outperform the state of the art. The work of Park in [18]

and the work of Di Lello et al. [12] most closely resemble our work. In Park et al.'s work, their multi-perceptron classifier classified 12 common anomalies with 90% accuracy. Furthermore, the paper also includes experiments where the robot feeds a real person with quadriplegia. In this work, they conducted anomaly identification and classification (they also classified the cause of the anomaly) and had 86% and 90% accuracy, resulting in a combined 88% effectiveness for the system. So with regards to anomaly classification, we still outperformed the accuracy marker, nonetheless the number of cases they considered was larger (12 instead of 5). With regards to the combined system, our (AD/AC) overall performance was of 94.62%, about 6% points higher than their, but again for a smaller number of anomaly cases. In Di Lello et al.'s work, they use a simple non-parametric Bayesian model, namely the sHDP-HMM with Gaussian observations and Gibb's sampling to classify anomalies. In their work, they achieved an average classification accuracy of 87.5% over four anomaly classes in an alignment skill with 4 obstructing objects. Our performance was between 6-8% points higher: 96.15% across nodes (Fig. 14 in Exp. 2) and 94.4% was the confusion matrix average in Exp. 2 (Fig. 16). Again, comparisons are difficult. Their experimentation consisted of single anomaly scenarios that did not change over time. Our scenarios included a wide range of anomalies, from one to multiple, occurring at different phases of the task with different objects and users. So, given that our anomaly experimentation was considerably more complex.

We would like to note the time and human cost of labelling anomalies in unstructured environments. The manual process was arduous and problematic and may not be reflective of a true warehouse scenario. Automating the anomaly label collection process through simulation or a farm of robots as in [49]) is possible, though the algorithm by which anomaly induction takes place should be examined to understand whether it approximates real-life conditions. Another possibility is the use of synthetically generated anomaly data. Synthetically generated data is becoming more common place [59–62], examples include synthetic voices, images, or depth representations. However, when it comes to anomaly data, the use of synthetic data seems more challenging as the structure of anomalous data can have important variations as discussed in this paper. It would be interesting to investigate the minimal amount of nominal data needed from which synthetic data could be generated with sufficient accuracy to properly introspect anomalies.

One more future line of research in anomaly classification is the ability to simultaneously identify multiple anomalies. Often times in our work human collisions resulted in object slips; however by simply selecting the class with

the highest likelihood, it is not possible to identify multiple anomalies simultaneously. We wish to explore this as a future line of work.

Finally, to further extend long-term autonomy horizons this work should be tested not just in isolated single tasks but in longer-term multi-task scenarios that can further test the effectiveness of the proposed approaches. Additionally, it would be interesting to consider more complex graph topologies in HRI, such as a dual-graph framework that synchronizes both human and robot activity and enables mutual introspection and recovery under explicit collaboration. We plan to extend our work to include hand-over tasks from humans to the robot instead of placing objects directly in the collection bin.

8 Conclusion

This work presented a tightly-integrated, graph-based online motion-generation, introspection, and incremental recovery system for manipulation tasks in loosely structured co-bot scenarios. Failures are and will continue to be a reality in robotics despite increasingly powerful motion-generation algorithms. Dealing with them explicitly has been the focus of this work. In this work we presented two recovery policies that allowed us to robustly recover from accidental and persistent anomalous conditions. Recovery ability grows in difficulty as the number of adaptations increases due to larger sensory-motor signal variations. On the other hand, the system self-repaired. On occasion, after an anomaly misclassification and improper recovery policy enactment, the system would correct its introspection and emit a successful recovery policy to complete the task. Ultimately, the system presented in this work significantly extended the autonomy and resilience of the robot and has broad applicability to all manipulation domains that suffer from uncertainties in unstructured environments.

Appendix A: Graph Structure

Nodes

In principle, a node specifies a motion-generation model and an associated goal. Two type of nodes are specified in our system: nominal and adaptive nodes.

Nominal Nodes

Nominal nodes are implemented as ROS-SMACH states whose class definition contains member functions. In

the specific case of DMPs, these are labeled as “get_dmp_model” and “get_pose_goal” for model and goal retrieval respectively. There is an additional attribute of integer type in the class definition acting as the ID of the nominal node.

Adaptive Nodes

Adaptive nodes are not implemented as a specific entity but rather as two procedures.

The first procedure concerns when and how to create an adaptive node. Since a new type of adaptation can only be brought into our system via human demonstration, we create a new adaptive node after a human demonstration has curred. The new adaptive node simply contains a unique integer as its ID and a DMP model trained from that human demonstration.

The second procedure concerns how to determine the goal for an adaptive node. If we were to use the last frame of a human demonstration as the goal, it would result in an adaptive node having little or no generalization ability due to the fixed structure. We thus propose that the goal of an adaption is a linear transformation with respect to the previous goal of the system. This linear transformation can be retrieved by computing the transformation matrix from the previous goal to the last frame of human demonstration. This information is then saved alongside the model of the adaptive node. At runtime, we can determine the skill goal of an adaptive node by applying the saved linear transformation on the previous goal of the system.

Node Transitions

Transitions Across Nominal Nodes

Since nominal nodes are implemented as SMACH states, we inherit SMACH’s state transition paradigm as our node transition paradigm. In the ROS-SMACH state definition, the member function named “determine_successor” is called by our system to determine a nominal node’s successor.

Transitions Among Adaptive Nodes

Since an adaptive node are entered only after an anomaly has occurred, we create a mapping from anomalies to their corresponding adaptive nodes. A key aspect of the mapping is a “compound key” composed of the ID of the node in which the anomaly happened and the anomaly type. For example, a key could be “nominal_node_(4)_anomaly_type_(tool_collision)”.

After an adaptive node terminates its motion, we must consider the successor node. The system assumes that adaptive nodes, perform recovery for a nominal node that previously failed and that must arrive at the next phase or milestone of the task. In this sense, when the adaptive node terminates, it signals that a nominal state into the next phase has been attained. In this case, the originally nominal node that experienced an anomalous condition should now regain its control in determining its successor such that the original task control flow could continue as if no anomaly happened at all.

Appendix B: Kitting Anomaly Dataset

The dataset captures sensory-motor and video data regarding the Kitting experiment under anomalous scenarios as outlined in this paper. The dataset consists of 538 rosbags. 85 of those rosbags are paired with RGB video that was captured by an external camera placed directly in front of the robot. The size of the 538 rosbags is of 37GB whilst the size of all videos is of 3.1GB. The dataset is found as Supplement 2 in the paper as well as in [63].

Data Description

The main content of our dataset is the sensory-motor recordings of the robot manipulator’s experience while performing the manipulation task. Specifically for the Rethink Baxter robot, we use the following data modalities:

- the right endpoint state: contains end-effector pose, twist, and a wrench defined from the joint torques (not used).
- the stamped wrench: obtained from a Robotiq FT 180 force-torque sensor installed on the right wrist (see Fig. 4).
- tactile data: obtained from a custom designed tactile sensor (see Section 8).

When anomalies are triggered, we also record: (i) the time-stamp at which the anomaly is flagged as well as the anomaly classification label.

Recording Methodology

All sensory-motor signals exist as ROS topics in our system and as such recorded as ROS bags offline. When an anomaly is identified, we signal this event by sending a timestamped ROS message to a pre-defined topic that is also recorded

as a rosbag. Anomaly classification labels are recorded in a text file in a line-by-line basis.

Mapping from data modalities to ROS topics is as follows:

- Baxter right endpoint state
/robot/limb/right/endpoint_state
- Robotiq force sensor FT 180
/robotiq_force_torque_wrench
- Robotiq tactile sensor
/TactileSensor4/Accelerometer,
/TactileSensor4/Dynamic,
/TactileSensor4/EulerAngle,
/TactileSensor4/Gyroscope,
/TactileSensor4/Magnetometer,
/TactileSensor4/StaticData

Data Organization

The dataset is composed of folders that use the format: "experiment_at.[time]". Each folder represents a test trial in the kitting experiment. Within a given folder, there will be a rosbag "record.bag" and a text file "anomaly_labels.txt". Each of these contain the rosbag topics mentioned in Section B and the recorded labels for the given experiment.

Anomaly Data Extraction

To extract anomaly data, one should first focus on the topic "/anomaly_detection_signal" whose messages are effectively timestamps indicating when anomalies were identified. It's worth noting that a burst of anomaly timestamps might have been published to this topic for one anomaly. Therefore timestamps that are adjacent in time should be ignored. We recommend ignoring a timestamp if its distance to its precursor is less than 1 second. After anomaly timestamps are extracted, labels in the accompanied "anomaly_labels.txt" can be paired accordingly.

We have tried to clear the dataset of any corrupted trials. However, if the number of anomaly timestamps does not equal to the number of labels, that experiment should be discarded.

Appendix C: Notation Table

Appendix D: Motor Skills

The DMP framework encodes dynamical systems through a set of nonlinear differential equations whose point attractor system is defined by a nonlinear forcing function, which in turn depends on a canonical system for temporal scaling. In this section, we introduce the main concepts and leave it to

the reader to refer to the original text for details. Formally, for a one DoF point attractor system, the point attractor system is defined as [64]:

$$\begin{aligned} \tau \dot{v} &= K(g - x) - Dv - K(g - x_0)s + Kf(s), \\ \tau \dot{x} &= v. \end{aligned} \tag{14}$$

Equation 14, is an extended PD control signal with spring and damping constants K and D respectively, position and velocity x and v , goal g , scaling s , and temporal scaling factor τ .

Table 10 Summary of graph and DMP notation

Notation	Description
Graph	
\mathcal{C}	Graph for a given task
\mathcal{B}	Behavior in a given task
\mathcal{N}	Behaviors are represented by nodes in the graph
\mathcal{T}	Transitions in a graph
$\mathcal{T}_{s,t}$	Node transitions from \mathcal{N}_s to \mathcal{N}_t
\mathcal{N}_{ij}	1st branch level node
\mathcal{N}_{ijk}	2nd branch level node
\mathcal{S}	Skill generation modules
\mathcal{V}	Visual goal processing modules
\mathcal{M}	Introspection modules
\mathcal{F}	A given anomaly
\mathcal{R}	A recovery action
\mathcal{R}_R	A re-enactment recovery type
\mathcal{R}_A	An adaptive recovery type
DMPs	
K	Spring constant of PD control
D	Damping constant of PD control
x, g, v	Position, position goal, & velocity
s	Spatial scaling constant
τ	Temporal scaling constant
α	Arbitrary scaling term
$f(\cdot)$	The forcing term
$\psi(\cdot)$	The basis function
ω	Weighting of basis functions
HMM	
z_t	Latent state at time t
\mathcal{X}_n	The n th training example sequence
x_t	Observation at time instant t
$b(z_t)$	Mode specific emission distribution
θ	Set of dynamic parameters of state k
π_0	Initial mode distribution
π_{jk}	Transition probability from state k to j
Π	A given HMM model
sHDP-HMM	
G_0	Base probability measure
G_j	HMM transition probability measure

Table 10 (continued)

Notation	Description
α, γ	DP concentration parameters
H	Continuous base distribution
β_k	Weights used to compute G_0
κ	The sticky parameter of transition distribution
$GEM(\gamma)$	Distribution to define stick-breaking process
Anomaly Identification	
∇L	Natural log of HMM filtered belief state
VAR	
e_t	Additive white noise at time t and mode z_t
Σ	White noise covariance matrix for mode z_t
A	Time-invariant regression matrix at z_t
Δ, ν	Covariance Δ & degrees of freedom ν in \mathcal{IW}
$\mathcal{IW}(\nu, \Delta)$	Inverse wishart
K	Covariance across matrix columns

The scaling term is controlled the canonical dynamical system $\tau \dot{s} = -\alpha s$, where α can be an arbitrary constant.

The forcing term $f(s)$ is an arbitrary function that, in our work, is provided by the user demonstration. The term is defined as a phase-dependent linear combination of Gaussian basis functions $\psi_i(s)$ with variable weights [64]. Spatio-temporal scaling is possible through the $(g - x)$ term in Eq. 14, which enables the system to adjust to varying goals. System speed-up is also possible through the τ variable in Eq. 14.

sHDP-AR-HMM Parameters & Hyperparameters

For the observation model, we use a first-order vector autoregressive with regression matrix coefficients A and covariance matrix Σ for specific latent states. Since both of these dynamic parameters are uncertain, they need to be learned. The MNIW is an appropriate prior distribution when both the mean and the covariance are uncertain [36].

We begin by determining the covariance Σ through the use of the IW distribution NIW . For this computation, we must define the first moment of the distribution according to Eq. 4. Here, we set ν , the degrees of freedom to the number of dimensions plus two: $\nu = d + 2$. This setting ensures the conjugate MNIW prior has a valid mean (see Sec. 4.5.1 in [42]). As for the computation of the expectation of the covariance in Eq. 5, the scalar s_F is set to 1.0 and multiplied by the scatter matrix (also the empirical covariance). This setting is motivated by the fact that the covariance is computed from polling all of the data and it tends to overestimate latent-state-specific covariances. A value slightly less than or equal to 1 of the constant in the scatter matrix mitigates the overestimation.

Then, to determine the matrix A of regression coefficients, the matrix normal of the MNIW uses a mean matrix M set to the zeros matrix $M = \mathbf{0}_d$, of size $d \times d$. We do so to let the new observation be primarily be determined by the signal noise.

For the covariance K across the columns an identity matrix is used such that $K = 1.0 * \mathbf{I}_d$ with the same dimension as Σ .

For the concentration parameter α of the HDP prior, a $Gamma(a, b)$ distribution with values $a = 0.5, b = 5$ is used. For the self-transition parameter μ a weakly informative $Beta(c, d)$ prior distribution is used with values $c = 1, d = 10$.

For the sticky HMM transition distribution, another κ (the degree of self-transition bias) is set to 50. The number of maximum iterations for the Split-Merge Monte Carlo method is set to 1000. Finally, the truncation (maximum) number for latent states is empirically set to $K = 10$ for both anomaly identification and classification.

Appendix E: Experimentation

Human Subject Training

In Exp. 3-6, five different human subjects, under consent, took part in the experiment as human collaborators. They were trained to place consumer goods, one-at-a-time, in the collection bin of the robot. We ask human subjects to assume they are multi-tasking and experiencing loss of attention. The loss of attention can lead (as recorded by the cataloging experiments in Section 2.2) to a number of anomalous events including: (i) HCs, (ii) TCs, (iii) OSs, and (IV) NOs—wall collisions (WC) are introduced in Exp. 4 but these are not caused by humans but from using DMPs that were trained with a particular geometry and size and testing with objects that differed from training. HCs may occur when the robot picks up objects from the collection bin and the human collaborator places new ones. TCs may occur when humans inadvertently place objects near each other such that when the robot attempts to pick an object, one of its fingers collides with the adjacent object (see Fig. 9b). OSs may occur after human collisions that rattle the gripper and cause heavier or smoother objects to fall. NO anomalies may occur when a human accidentally collides or removes an object that the robot intended to pick up.

Signal Processing

Observations consists of a 7 DoF pose (using quaternions as orientation), a 6 DoF end-effector twist and wrench, and 56 taxel values (each finger has a 4-by-7 grid). Various pre-processing techniques were tested for a combination

of these features. We conducted validation to select the optimal feature set. Details for Anomaly Identification and Classification are reported in Exp. 1 and 2 respectively.

All signals were scaled, resampled, and aligned.

Signals were scaled to lie in a range of $-1 \leq y_i \leq 1$ by computing the absolute value of the maximum signals during training. Different signals publish at different rates (wrench: 1000Hz, tactile: 1000Hz, pose and twist: 100Hz). We resample to acquire a single time-point to model the observations. Our code relies primarily on python and ROS. Rospay nodes inherently use Python's multi-threading class to handle multiple publishers and subscribers. The class, however, lacks real-time performance support and we have only achieved re-sampling rates of up to 50Hz. Alignment takes places by syncing the timestamps from the varying ROS topics.

Anomaly Identification Baselines

The HMM models an empirical covariance matrix with two observation models (Gaussian 'G' and Autoregressive 'AR') and two inference algorithms (Expectation-Maximization 'EM', Variational Bayes 'VB'). We use 3 different values for the complexity k of the HMM (3,5,10).

For machine learning, the Isolation Forest uses default values from sklearn, the maximum number of samples is set to automatic, and the contamination value set to 0.01. For LOF, default values are used. Exceptions are novelty set to true and contamination set to 0.01. The MLP and LSTM networks both use feature distribution #14, a batch size of 16, learning rate of 0.0005, a leaky relu fixed $\alpha = 0.2$ and an outlier fraction of 0.2. For the MLP, we use 18 input dimensions, 128 hidden units, and for the VAE latent states we use 16 dimensions. For the LSTM-VAE, we have a 16 time-steps input and 64 hidden states.

Exp. 4c

In Exp. 4c (Section 6.5.4), it was noted that one set of objects in particular posed challenges to the classification system. Under perfect classification, an adaptive behavior rotated the gripped object and cause a collision with objects leading to an irrecoverable situation. For imperfect classification, there was a set of trials that led to 0 completions. Failure occurred during the adaptation to the persistent wall collision in node 3 as the system moved to the box. The culprit was the inability of the system to adapt its motion when an object with different shape attributes (height) was used compared to the one used during user demonstrations. This result points to a weakness in the system's ability to generalize adaptations when object shapes vary drastically from training as no spatial reasoning is yet embedded in the system.

Exp. 6

In Exp. 6, there were three experiments that failed due to the following situation: during the 2nd adaptation attempt to grasp the block, the approach pose was inaccurate. Normally, our fingers open when a pre-pick motion has terminated. The approach trajectory had some imprecision and led to the fingers making contact with the block causing it to tip (instead of sliding along the block to reach an optimal pick pose). After the tip, the block was displaced beyond the field-of-view of the camera. At this point the system continued to *correctly trigger an NO flag*, however on re-enactment the pose of the object was unavailable; thus holding-up the execution of the re-enactment. This could be prevented by a better implementation of the manipulation skills taught to pick the object. In retrospect, we never envisioned that training the pick in this way would be problematic. It is unclear whether end-to-end training would suffer a similar problem from inception. Clearly, the adaptations could be re-trained or improved to address the issue under any manipulation scheme. The question remains as to which approach would be more robust to previously unseen situations.

Acknowledgments We would like to thank Prof. Vincent Duchaine with the Department of Automated Manufacturing Engineering at Quebec University for his kind support in donating the multimodal tactile sensor used in this work [31].

Author Contributions Shuangqi Luo did a large portion of the code development for the graph, anomaly identification, and testing along with ideas for all aspects of the work. Hongmin Wu strongly contributed in all areas of anomaly identification, classification, and experimentation. Shuangda Duan helped with manipulation coding, data collection, and experimentation. Yijiong Lin helped with performance comparisons and ablation studies. Juan Rojas provided the main ideas for the work as well as guidance for the team.

Funding This work is supported by grants from the Major Project of the Guangdong Province Department for Science and Technology [2019A050510040] and NSFC [61950410758], as well as the VC fund of the CUHK T Stone Robotics Institute (4930745).

Data Availability An extensive dataset has been collected for this work as described in Appendix B and is available via our project page www.JuanRojas.net/spair or at Github https://github.com/birlrobotics/ktting_anomaly_dataset. The authors of this paper have made all aspects of the coding of this work open-source and are well documented, please refer to our supplementary page [26].

Declarations

Ethics approval This study did not require ethics approval as the information consists of naturalistic observations regarding choice. Choices remained anonymous. There is not any identifier information that would allow attribution of private information to an individual.

Consent to participate Informed consent was obtained from all participants in the study.

Consent to publish Participant consented to the submission of this article to the journal.

Conflict of Interests All the authors of this paper have no conflicts of interest, financial or otherwise. The funding listed above poses no conflict to this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput.* **25**(2), 328–373 (2013)
- Paraschos, A., Daniel, C., Peters, J.R., Neumann, G.: Probabilistic movement primitives. In: *Advances in Neural Information Processing Systems*, pp. 2616–2624 (2013)
- Calinon, S., D'Halluin, F., Sauser, E.L., Caldwell, D.G., Billard, A.G.: Learning and reproduction of gestures by imitation. *IEEE Robot. Autom. Magazine* **17**(2), 44–54 (2010)
- Jain, A., Wojcik, B., Joachims, T., Saxena, A.: Learning trajectory preferences for manipulators via iterative improvement. In: *Advances in Neural Information Processing Systems* (2013). [Online]. Available: <http://pr.cs.cornell.edu/coactive>
- Konidaris, G., Kuindersma, S., Grupen, R., Barto, A.: Robot learning from demonstration by constructing skill trees. *Int. J. Robot. Res.* **31**(3), 360–375 (2012)
- Gutierrez, R.A., Chu, V., Thomaz, A.L., Niekum, S.: Incremental task modification via corrective demonstrations. In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1126–1133 (2018)
- Bajcsy, A., Losey, D.P., O'Malley, M.K., Dragan, A.D.: Learning from physical human corrections, one feature at a time. In: *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 141–149 (2018)
- Hovland, G.E., McCarragher, B.J.: Hidden Markov models as a process monitor in robotic assembly. *Model. Identif. Control* **20**(4), 201–223 (1999)
- Pettersson, O.: Execution monitoring in robotics: a survey. *Robot. Auton. Syst.* **53**(2), 73–88 (2005)
- Kobayashi, Y., Matsumoto, T., Takano, W., Wollherr, D., Gabler, V.: Motion recognition by natural language including success and failure of tasks for co-working robot with human. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. Institute of Electrical and Electronics Engineers Inc., pp. 10–15 (2017)
- Inceoglu, A., Ince, G., Yaslan, Y., Sariel, S.: Failure detection using proprioceptive, auditory and visual modalities. In: *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., pp. 2491–2496 (2018)
- Di Lello, E., Klotzbucher, M., De Laet, T., Bruyninckx, H.: Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5827–5833. IEEE (2013)
- Cheng, X., Jia, Z., Mason, M.T.: Data-efficient process monitoring and failure detection for robust robotic screwdriving. In: *IEEE International Conference on Automation Science and Engineering*, vol. 2019-Augus. IEEE Computer Society, pp. 1705–1711 (2019)
- Wu, H., Guan, Y., Rojas, J.: A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. *Appl. Sci. (Switzerland)* **9**(6), 1072 (2019). [Online]. Available: <https://www.mdpi.com/2076-3417/9/6/1072>
- Park, D., Erickson, Z., Bhattacharjee, T., Kemp, C.C.: Multimodal execution monitoring for anomaly detection during robot manipulation. In: *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 407–414 (2016)
- Park, D., Kim, H., Kemp, C.C.: Multimodal anomaly detection for assistive robots. *Autonomous Robots* **43**(3), 611–629 (2019). [Online]. Available: <https://doi.org/10.1007/s10514-018-9733-6>
- Luo, S., Wu, H., Lin, H., Duan, S., Guan, Y., Rojas, J.: Fast, robust, and versatile event detection through HMM belief state gradient measures. In: *The 27th IEEE International Symposium on Robot and Human Interactive Communication*, 2018. Proceedings. ROMAN 2018, vol. 2018-Janua. Nanjing, China: Institute of Electrical and Electronics Engineers Inc., pp. 1–8 (2018). [Online]. Available: <http://arxiv.org/abs/1709.07876>
- Park, D., Kim, H., Hoshi, Y., Erickson, Z., Kapusta, A., Kemp, C.C.: A multimodal execution monitor with anomaly classification for robot-assisted feeding. In: *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, pp. 5406–5413 (2017)
- Rodriguez, A., Mason, M.T., Srinivasa, S.S., Bernstein, M., Zirbel, A.: Abort and retry in grasping. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1804–1810. IEEE (2011)
- Wu, H., Luo, S., Lin, H., Duan, S., Guan, Y., Rojas, J., Luo, S., Duan, S., Guan, Y., Rojas, J.: Recovering from external disturbances in online manipulation through state-dependent reverte recovery policies. In: *RO-MAN 2018 - 27th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 166–173 (2018)
- Chang, G., Kulic, D., Kulić, D., Kulic, D.: Robot task error recovery using Petri nets learned from demonstration. In: *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6. IEEE (2013)
- Kappler, D., Pastor, P., Kalakrishnan, M., Wüthrich, M., Schaal, S.: Data-driven online decision making for autonomous manipulation. In: *Robotics: Science and Systems*, vol. 11 (2015). Rome, Italy
- Niekum, S., Osentoski, S., Konidaris, G., Chitta, S., Marthi, B., Barto, A.G.: Learning grounded finite-state representations from unstructured demonstrations. *Int. J. Robot. Res.* **34**(2), 131–157 (2015)
- Wang, A.S., Kroemer, O.: Learning robust manipulation strategies with multimodal state transition models and recovery heuristics. In: *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 1309–1315 (2019). [Online]. Available: https://www.ri.cmu.edu/wp-content/uploads/2019/03/Kroemer_Wang_ICRA_2019.pdf
- Wu, H., Lin, H., Guan, Y., Harada, K., Rojas, J., Wu, H., Lin, H., Guan, Y., Harada, K., Rojas, J.: Robot introspection with Bayesian nonparametric vector autoregressive hidden Markov models. In: *IEEE-RAS International Conference on Humanoid*

- Robots, vol. Part F1341, no. Nips. IEEE, pp. 882–888 (2017). [Online]. Available: <http://www.juanrojas.net/shdp-var-hmm/>
26. Wu, H., Luo, S., Chen, L., Duan, S., Chumkamon, S., Liu, D., Guan, Y., Rojas, J.: Endowing robots with longer-term autonomy by recovering from external disturbances in manipulation through grounded anomaly classification and recovery policies (2018). [Online]. Available: <http://www.juanrojas.net/spair>
 27. Kroemer, O., Daniel, C., Neumann, G., van Hoof, H., Peters, J.: Towards learning hierarchical skills for multi-phase manipulation tasks. In: International Conference on Robotics and Automation (ICRA), vol. 2015-June, no. June, pp. 1503–1510 (2015)
 28. Rojas, J., Luo, S., Zhu, D., Du, Y., Lin, H., Huang, Z., Kuang, W., Harada, K.: Online robot introspection via wrench-based action grammars. In: IEEE International Conference on Intelligent Robots and Systems, vol. 2017-Sept, pp. 5429–5436 (2017). [Online]. Available: http://www.juanrojas.net/online_introspection_wrench_grammar/
 29. Lin, H.C., Shafran, I., Yuh, D., Hager, G.D.: Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions. *Comput. Aided Surg.* **11**(5), 220–230 (2006)
 30. Rosen, J., Brown, J.D., Chang, L., Sinanan, M.N., Hannaford, B.: Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete Markov model. *IEEE Trans. Biomed. Eng.* **53**(3), 399–413 (2006)
 31. Le, T.H.L., Maslyczyk, A., Roberge, J.P., Duchaine, V.: A highly sensitive multimodal capacitive tactile sensor. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 407–412. IEEE (2017)
 32. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: Springer Tracts in Advanced Robotics, vol. 15, pp. 561–572. Springer (2005)
 33. Grollman, D.H., Jenkins, O.C.: Incremental learning of subtasks from unsegmented demonstration. In: IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings. IEEE, pp. 261–266 (2010)
 34. Rojas, J., Peters Ii, R.A., Peters, R.A., Peters Ii, R.A., Peters, R.A.: Sensory integration with articulated motion on a humanoid robot. *Appl. Bionics Biomechan.* **2**(3–4), 171–178 (2005)
 35. Fox, E.B., Sudderth, E.B., Jordan, M.I., Willsky, A.S.: Bayesian nonparametric methods for learning markov switching processes. *IEEE Signal Process. Mag.* **27**(6), 43–54 (2010)
 36. Hughes, M.C., Stephenson, W.T., Sudderth, E.B.: Scalable adaptation of state complexity for nonparametric hidden Markov models. *Adv. Neural Inform. Process. Syst.* **2015-Janua**, 1198–1206 (2015)
 37. Fox, E.B., Hughes, M.C., Sudderth, E.B., Jordan, M.I., et al.: Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Ann. Appl. Stat.* **8**(3), 1281–1313 (2014)
 38. Johnson, M.J., Willsky, A.S.: Stochastic variational inference for Bayesian time series models. In: 31st International Conference on Machine Learning, ICML 2014, vol. 5, pp. 3872–3880 (2014)
 39. Foti, N.N.J., Xu, J., Laird, D., Fox, E.B.: Stochastic variational inference for hidden Markov models. In: Advances in Neural Information Processing Systems, vol. 4, no. January, pp. 3599–3607 (2014)
 40. Chang, J., Fisher, J.W.: Parallel sampling of HDPs using sub-cluster splits. In: Advances in Neural Information Processing Systems, vol. 1, no. January, pp. 235–243 (2014)
 41. Bnpy: Bayesian nonparametric machine learning for Python. [Online]. Available: <https://github.com/bnpy/bnpy/> (2017)
 42. Murphy, K.P., Robert, C. In: Dietterich, T. (ed.): Machine Learning: a Probabilistic Perspective, vol. 27. MIT Press, Cambridge (2012)
 43. Nakamura, A., Nagata, K., Harada, K., Yamanobe, N., Tsuji, T., Foissotte, T., Kawai, Y.: Error recovery using task stratification and error classification for manipulation robots in various fields. In: IEEE International Conference on Intelligent Robots and Systems, pp. 3535–3542. IEEE (2013)
 44. Council, N.R., et al.: Modeling human and organizational behavior: application to military simulations. National Academies Press (1998)
 45. An, J., Cho, S.: SNU data mining center 2015-2 special lecture on IE variational autoencoder based anomaly detection using reconstruction probability, Seoul National University, Tech. Rep. (2015)
 46. Park, D., Hoshi, Y., Kemp, C.C.: A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder, vol. 3, pp. 1544–1551 (2018)
 47. Chen, R.-Q., Shi, G.-H., Zhao, W.-L., Liang, C.-H.: Sequential VAE-LSTM for anomaly detection on time series. [Online]. Available: <http://arxiv.org/abs/1910.03818> (2019)
 48. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. *J. Machine Learn. Res.* **17**(1), 1334–1373 (2016). [Online]. Available: <http://www.jmlr.org/papers/volume17/15-522/15-522.pdf>
 49. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **37**(4–5), 173–184 (2016)
 50. Haarnoja, T., Pong, V., Zhou, A., Dalal, M., Abbeel, P., Levine, S.: Composable deep reinforcement learning for robotic manipulation. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 6244–6251 (2018)
 51. Jund, P., Eitel, A., Abdo, N., Burgard, W., Philipp Jund Andreas Eitel, N.A., Burgard, W.: Optimization beyond the convolution: generalizing spatial relations with end-to-end metric learning. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 4510–4516 (2018)
 52. Adjali, O., Ramdane-Cherif, A.: High-level MLN-based approach for spatial context disambiguation. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 2909–2915 (2018)
 53. Aly, A., Taniguchi, T.: Towards understanding object-directed actions: a generative model for grounding syntactic categories of speech through visual perception. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 7143–7150 (2018)
 54. Gong, Z., Zhang, Y.: Temporal spatial inverse semantics for robots communicating with humans. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 4451–4458 (2018)
 55. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from RGB-D videos. *Int. J. Robot. Res.* **32**(8), 951–970 (2013). [Online]. Available: <http://dx.doi.org/10.1177/0278364913478446>
 56. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. *IEEE Trans. Pattern Anal. Machine Intell.* **38**(1), 14–29 (2016)
 57. Paulius, D., Huang, Y., Milton, R., Buchanan, W.D., Sam, J., Sun, Y.: Functional object-oriented network for manipulation learning. In: IEEE International Conference on Intelligent Robots and Systems, vol. 2016-Novem. IEEE, pp. 2655–2662 (2016)
 58. Jelodar, A.B., Sirajus Salekin, M., Sun, Y.: Identifying object states in cooking-related images, arXiv. [Online]. Available: <http://arxiv.org/abs/1805.06956> (2018)
 59. Radovanov, B., Marcikić, A., Larrue, D., Legeard, M.: A comparison of four different lens mappers. *Croatian Oper. Res. Rev.* **91**(2), 189–202 (2014)

60. Forestier, G., Petitjean, F., Dau, H.A., Webb, G.I., Keogh, E.: Generating synthetic time series to augment sparse datasets. In: Proceedings - IEEE International Conference on Data Mining, ICDM, vol. 2017-Novem. IEEE, pp. 865–870 (2017)
61. Vinod, H.D., López-de Lacalle, J.: Others Maximum entropy bootstrap for time series: the meboot R package. *J. Stat. Softw.* **29**(5), 1–19 (2009)
62. Guennec, A.L., Malinowski, S., Tavenard, R., Le Guennec, A., Malinowski, S., Tavenard, R.: Data augmentation for time series classification using convolutional neural networks. In: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data (2016)
63. Wu, H., Luo, S., Chen, L., Duan, S., Chumkamon, S., Liu, D., Guan, Y., Rojas, J., Wu, H., Luo, S., Chen, L., Duan, S., Chumkamon, S., Liu, D., Guan, Y., Rojas, J.: Endowing robots with longer-term autonomy by recovering from external disturbances in manipulation through grounded anomaly classification and recovery policies, Arxiv (2018). [Online]. Available: <http://arxiv.org/abs/1809.03979>http://www.juanrojas.net/re_enact_adapt/
64. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: IEEE International Conference on Robotics and Automation, 2009. ICRA'09, vol. 2009, pp. 763–768. IEEE (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shuangqi Luo received his B.E. degree in Software Engineering from Sun Yat-Sen University, Guangzhou, China, in 2017. Later he served as a research associate at the Guangdong University of Technology. He is currently working towards his Ph.D. degree in the University of Maryland, College Park, USA. His research interests include human-like robot control, cognitive robotics, and philosophy of mind.

Hongmin Wu is a post-doctoral research fellow at the Guangdong Institute of Intelligent Manufacturing in the Guangdong Key Lab of Modern Control Technology. He completed his Ph.D. in Mechanical Engineering at the Guangdong University of Technology, Guangzhou, China, in 2019. His research interests lie in the domain of multimodal time series modeling using Bayesian nonparametric methods, robot introspection, movement representation as well as robot learning.

Shuangda Duan received the B.S. degree in Mechanical and Energy Engineering from Shaoyang University, Hunan, China, in 2012, and a M.S. degree in Mechanical Engineering from Guangdong University of Technology, Guangzhou, China, in 2018. He is currently working as a project manager at the University of Waterloo (Foshan) Innovation Center.

Yijiong Lin is currently a Master's degree candidate. He received his B.S. degree in Mechanical Engineering from the Guangdong University of Technology in 2017. His research interests include reinforcement learning in robotics.

Juan Rojas received his B.S., M.S., and Ph.D. in Electrical and Computer Eng from Vanderbilt University in 2002, 2004, and 2009 respectively. He is an Assistant Research Professor at the Mechanical and Automation Engineering Department at the Chinese University of Hong Kong. His research interests include intelligent manipulation skills, introspection, and decision making. Dr. Rojas is an IEEE Senior Member.