Multimodal Anomaly Detection with a Mixture-of-Experts

Christoph Willibald*,1 and Daniel Sliwowski*,2 and Dongheui Lee[1,2]

***Abstract*— With a growing number of robots being deployed across diverse applications, robust multimodal anomaly detection becomes increasingly important. In robotic manipulation, failures typically arise from (1) robot-driven anomalies due to an insufficient task model or hardware limitations, and (2) environment-driven anomalies caused by dynamic environmental changes or external interferences. Conventional anomaly detection methods focus either on the first by low-level statistical modeling of proprioceptive signals or the second by deep learning-based visual environment observation, each with different computational and training data requirements. To effectively capture anomalies from both sources, we propose a mixture-of-experts framework that integrates the complementary detection mechanisms with a visual-language model for environment monitoring and a Gaussian-mixture regression-based detector for tracking deviations in interaction forces and robot motions. We introduce a confidence-based fusion mechanism that dynamically selects the most reliable detector for each situation. We evaluate our approach on both household and industrial tasks using two robotic systems, demonstrating a 60% reduction in detection delay while improving frame-wise anomaly detection performance compared to individual detectors.**

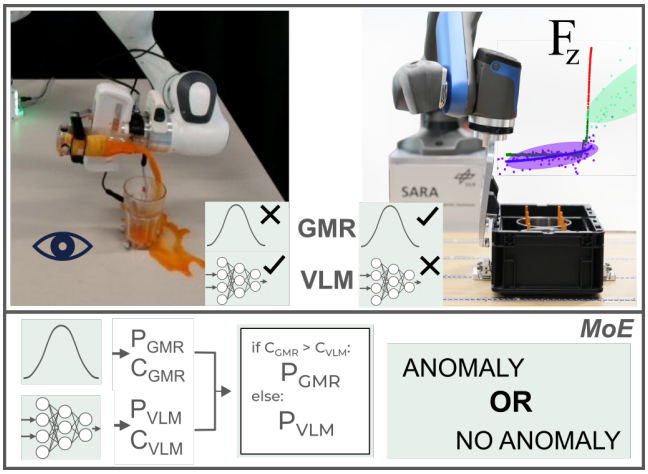


Fig. 1: Our Mixture of Experts (MoE) framework that combines GMR-based outlier detection with VLM-based scene monitoring to identify anomalies in both household and industrial settings. The framework dynamically selects the most reliable detector for each situation based on a prediction confidence score.

I. Introduction

As collaborative robots act increasingly autonomously across various applications, accurately monitoring task progress and success becomes crucial. In both household and industrial settings, like the ones depicted in Fig. 1, autonomous robots are confronted with a range of unknown situations, leading to diverse sources of task failure. Common anomalies arise from hardware defects, human interference, or inaccuracies in the task model. To account for those diverse failure cases, an anomaly detection mechanism must integrate multimodal sensor data to decide whether the current situation constitutes an anomaly.

Conventional anomaly detectors in robotics, whether online or offline, typically follow two approaches: modeling the statistical distribution of low-dimensional data to identify outliers or leveraging deep neural networks to detect anomalies from high-dimensional inputs like video [1]. Approaches of the former category [2]–[7] perform well in unsupervised detection of deviations from proprioceptive measurements with sparse training data. For that, probabilistic distributions of expected measurements are learned from successful executions to detect deviations such as anomalous process forces with different thresholding methods. Deep learning-based detectors [8]–[17] utilize a supervised of self-supervised setting, typically requiring large training datasets of both positive and negative task examples, which makes them more effective at recognizing failures in environmental interactions, such as spilling liquid during pouring. Since the different detection methods vary in their training data requirements and computational complexity, it is beneficial to train individual detectors on distinct data subsets. However, to robustly detect anomalies in robotic manipulation tasks, a combined identification of deviations in motion, force, and interaction dynamics between the robot and its environment is required. Therefore, a comprehensive anomaly detection system must integrate multiple specialized subdetectors into a unified framework.

We propose a novel multimodal anomaly detection strategy for robust task monitoring, integrating multiple detection approaches within a mixture-of-experts framework. Our method combines two complementary detectors: one analyzing the robot's low-level dynamics and the other visually monitoring the scene to detect deviations from the expected task execution. A visual-language model (VLM) [8] detects violations of action preconditions and effects resulting from interaction with the environment, while a Gaussian-mixture

⋆ Equal contribution.

[1]Institute of Robotics and Mechatronics (DLR), German Aerospace Center, Wessling, Germany.

[2]Autonomous Systems Lab, Institute of Computer Technology, TU Wien, Vienna, Austria.

This work has been partially supported by the European Union project INVERSE under grant agreement No. 101136067, partially by the DLR internal project ASPIRO, and in part by the Robot Industry Core Technology Development Program under Grant 00416440 funded by the Korea Ministry of Trade, Industry and Energy (MOTIE).

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: 10.1109/IROS60139.2025.11245878

regression (GMR)-based detector [2] continuously tracks deviations in interaction forces and robot poses. Considering the prediction confidence of each approach, their outputs are fused to produce a joint result. This late fusion approach allows the framework to dynamically select the most suitable detector in a given situation. Figure 1 highlights scenarios where either VLM-based or GMR-based anomaly detection is most effective, emphasizing the versatility and advantages of a fused approach.

The contributions of our work are as follows:

1) a visual-language anomaly detection approach based on [8] which uses task progress to ground the expected action state and obtain a prediction confidence score;
2) a probabilistic anomaly detection method based on [2] using local modality-specific anomaly thresholds and quantification of prediction confidence;
3) a mixture-of-experts fusion strategy for merging probabilistic and deep-learning-based anomaly detection approaches.

II. Related works

Previous works in robotics attempt to solve the anomaly detection problem by using either probabilistic modeling or deep learning.

A. Probabilistic anomaly detection

Probabilistic approaches in [2]–[4] model successful task executions with Gaussian Mixture Models (GMM). By regressing the model on time, these methods compute expected sensor measurements and dynamically adjust anomaly detection sensitivity through probabilistic modeling. Chernova et al. [18] employ a GMM to encode a simple policy using basic symbolic actions and utilize the observation likelihood of unseen states for outlier detection. An offline anomaly detection method is presented in [5], where Gaussian Processes Regression is used to predict force profiles and epistemic uncertainty during insertion tasks to compute an anomaly score at every time step. A run is unsuccessful if the average anomaly score over all time steps exceeds a predefined threshold. Probabilistic detection of anomalies can also be achieved with a Hidden Markov Model, either through predefined anomaly thresholds combined with a sliding time window approach [7], or by estimating probabilistic thresholds based on execution progress [6]. These methods treat anomaly detection as statistical outlier detection of low-dimensional features derived from multimodal sensor data. However, unlike our VLM-based detector, they do not incorporate visual observations of the scene to identify anomalous effects of the robot's actions on the environment.

In [19], the action consistency of a diffusion policy is evaluated across consecutive timesteps using a statistical distance function. An anomaly is detected when the cumulative sum of these distances exceeds a threshold learned from successful executions. This detector is combined with an LVLM to assess task progress through chain-of-thought reasoning and video question-answering, determining whether the robot is still progressing toward the task goal. However, unlike our approach, this method cannot detect deviations in contact force and exhibits significantly longer response times for the LVLM-based detector.

B. Deep-Learning anomaly detection

Probabilistic approaches perform well with low-dimensional data, such as end-effector poses or force-torque measurements, but struggle with high-dimensional data like images or audio [1]. Deep-learning methods address this by learning feature representations while detecting anomalies [9]–[12]. Broadly, deep-learning anomaly detection techniques fall into four categories: reconstruction-based approaches, one-class neural networks, success detectors, and variants of visual- and large language models.

Reconstruction-based methods treat anomaly detection as a self-supervised task, training generative models such as autoencoders [20], variational autoencoders [9], or generative adversarial networks [21] to encode and decode observations. Since they are trained only on "normal" data, they struggle to reconstruct anomalies, which results in a higher reconstruction loss. Anomalies are detected by thresholding this loss. These methods have been successfully applied in robotics tasks, such as autonomous feeding [9]. One-class neural networks extend the one-class support vector machine to deep architectures [13]. They introduce loss functions that separate normal data representations using a hyperplane or hypersphere in latent space. Instead of reconstruction loss, they produce an "outlier score" that is thresholded to detect anomalies.

Success detection approaches are specific to robotic task execution anomaly detection, framing anomaly detection as a supervised classification problem. A neural network classifies executions as successful or unsuccessful based on single or sequential observations from the execution. These methods require both positive and negative task demonstrations. Vision-language models like SuccessVQA [14] apply this concept, while FinoNET [15] integrates video and audio data to make the predictions.

Lastly, a range of deep learning approaches combine vision and language to enable reasoning-based anomaly detection. Many of these methods leverage large language models (LLMs) or large vision-language models (LVLMs) to assess task progress and identify unexpected behavior. For example, TP-VQA [16] queries an LVLM using predicates from a task's PDDL [22] description to check whether the expected conditions hold. Similarly, AESOP [17] compares text embeddings of the current and reference scene descriptions, using an LLM to determine whether the difference is significant. AHA-VLM [23] extends this idea by fine-tuning an LVLM to both detect anomalies and reason about their causes.

In contrast to these LLM-based methods, ConditionNet [8] learns action preconditions and effects directly from data, identifying anomalies by comparing predicted and expected action states. Another approach, proposed by Xu et al. [24], avoids the need for anomaly examples entirely. Their method

trains only on successful executions by transforming visual and proprioceptive inputs into a latent noise space using normalizing flows, and detects anomalies based on the magnitude of the resulting noise vectors.

Deep-learning approaches also allow leveraging of multimodal information, by appropriately designing the network architecture. Works like [10]–[12] adopt a late fusion strategy, where features from each modality are extracted separately and then concatenated and classified with a Multi-Layer perception. Alternatively, works like [9] adopt an early fusion strategy, where raw sensory readings are concatenated, and the model learns to reconstruct the data for successful executions.

In this work, we propose a mixture of experts that uses a late fusion strategy to merge the predictions from a probabilistic model (GMM) and a visual-language anomaly detector (VLM) [8]. The VLM detects anomalies on a semantic level, e.g. a spill is when liquid is outside of the cup, while the GMM encodes the expected low-level dynamics of the task including process forces, and detects deviations from it. As a result, the individual detectors of our method are trained on different training data, which improves the performance of each detector.

III. Methodology

A. Problem Formulation

To avoid and react to errors during autonomous task execution, a robot must be able to identify unintended or faulty situations. Our proposed multimodal approach leverages proprioceptive and exteroceptive sensor measurements to achieve robust anomaly detection performance. Proprioception refers to contact force and end-effector pose collected with the robot, whereas exteroception includes visual features and object poses in the environment. Our proposed anomaly detection approach combines a probabilistic GMR-based detector $GMR_\theta(\boldsymbol{\xi}_t)$ with a visual-language classification-based action monitoring approach $VLM_\psi(F_t, a^t_{nl})$. An overview of the mixture-of-experts anomaly detection pipeline can be seen in Figure 2.

Given a dataset $\mathscr{D} = \{\boldsymbol{d}_i\}_{i=1}^{N_{total}}$ of N_{total} demonstrations, where each demonstration $\boldsymbol{d}_i$ consists of proprioceptive and exteroceptive sensory readings $\boldsymbol{p}_i = \{\boldsymbol{p}_i^n\}_{n=1}^{N_i}$ and $\boldsymbol{e}_i = \{\boldsymbol{e}_i^n\}_{n=1}^{N_i}$ of length N_i, a natural language task description a^i_{nl}, and the success status $success_i$, i.e., $\boldsymbol{d}_i = (\boldsymbol{p}_i, \boldsymbol{e}_i, a^i_{nl}, success_i)$, the goal is to learn two anomaly detection models, $GMR_\theta(\boldsymbol{\xi}_t)$ and $VLM_\psi(F_t, a^t_{nl})$, which are fused into a mixture-of-experts model $MOE_{\theta,\psi}(\boldsymbol{o}_t)$, where $\boldsymbol{o}_t = (\boldsymbol{\xi}_t, F_t, a^t_{nl})$ is the current observation during the execution, containing low-level features $\boldsymbol{\xi}_t$, the current video frame F_t, and the natural language task description. The parameter set θ includes the GMM's means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$, mixing coefficients π_k, and anomaly thresholds $\boldsymbol{D}^k_{M,max}$ for each of the K components: $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, \boldsymbol{D}^k_{M,max}\}_{k=1}^K$. The set ψ contains the weights and biases of the transformer-based vision-language anomaly detection model [8]. To fuse the results from the GMR-based expert and the VLM-based expert, given $\boldsymbol{o}_t$, the mixture of experts compares the confidence score C and anomaly prediction P of both models, denoted with subscripts GMR and VLM.

To obtain the dataset $\mathscr{D}$, various demonstration techniques such as kinesthetic teaching [25], haptic teleoperation [26], and VR teleoperation [8] can be used. Different experts in the mixture-of-experts model focus on different aspects of the data. For instance, since the GMM anomaly detection model relies on force and torque information, it is crucial that demonstrations capture accurate force and torque profiles. Thus, kinesthetic teaching is the most suitable technique. Conversely, the VLM anomaly detection model leverages only visual data, making it essential that the dataset accurately represents scene states during action execution. If the autonomous execution mode rarely includes a visible human, kinesthetic teaching becomes unsuitable, as it would always include the demonstrator. In such cases, VR or haptic teleoperation is preferable. Moreover, different experts may require varying amounts of data to model anomalies effectively. To ensure optimal training, we divide the dataset into two subsets: $\mathscr{D} = \mathscr{D}_{tele} \cup \mathscr{D}_{kin}$, where $\mathscr{D}_{tele} = \{d_i\}_{i=0}^{N_{tele}}$ is collected via teleoperation and $\mathscr{D}_{kin} = \{d_i\}_{i=0}^{N_{kin}}$ via kinesthetic teaching.

B. Probabilistic anomaly detection

For each skill, a probabilistic model of expected feature values is learned from kinesthetic user demonstrations. The feature values $\boldsymbol{\xi}$, including contact forces, relative distances, and orientations between the end-effector and objects during manipulation, and are computed from the proprioceptive and exteroceptive sensor readings collected during N_{kin} task demonstrations of length H_n. The obtained training set containing the L relevant features for a skill

$$\{\{\boldsymbol{\xi}_{h,n}\}_{h=1}^{H_n}\}_{n=1}^{N_{kin}}, \boldsymbol{\xi}_{h,n} \in \mathbb{R}^L \tag{1}$$

is encoded as a Gaussian Mixture Model (GMM), estimating the joint probability distribution of the training data set as a weighted sum of K independent Gaussian components

$$\boldsymbol{\xi} \sim \sum_{k=1}^{K} \pi_k \mathscr{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

where $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ are the mixing coefficient, feature mean, and covariance matrix of the kth Gaussian component. Similar to our previous work [2], [3], we can decompose the features into an input and output set $\boldsymbol{\xi} = [\boldsymbol{\xi}^{\mathrm{I}}, \boldsymbol{\xi}^{\mathrm{O}}]^{\mathrm{T}}$. The choice of features, as well as their division into input and output components, depends on the specific characteristics of the task at hand. A commonly used configuration considers the end-effector pose relative to the manipulated object as the input, and the contact forces as the output. We use Gaussian Mixture Regression (GMR) at every time step t during the task execution to infer the expected output feature vector $\mathbb{E}(\boldsymbol{\xi}^{\mathrm{O}}_t | \boldsymbol{\xi}^{\mathrm{I}}_{t,\mathrm{M}}) = \hat{\boldsymbol{\mu}}^{\mathrm{O}}_t$ along with the covariance matrix $\hat{\boldsymbol{\Sigma}}^{\mathrm{O}}_t$ given the measured input feature vector $\boldsymbol{\xi}^{\mathrm{I}}_{t,\mathrm{M}}$ by computing the conditional probability distribution $P(\boldsymbol{\xi}^{\mathrm{O}}_t | \boldsymbol{\xi}^{\mathrm{I}}_{t,\mathrm{M}}) = \mathscr{N}(\boldsymbol{\xi}^{\mathrm{O}}_t | \hat{\boldsymbol{\mu}}^{\mathrm{O}}_t, \hat{\boldsymbol{\Sigma}}^{\mathrm{O}}_t)$. Using the measured output feature vector $\boldsymbol{\xi}^{\mathrm{O}}_{t,\mathrm{M}}$ at time t, we compute the Mahalanobis

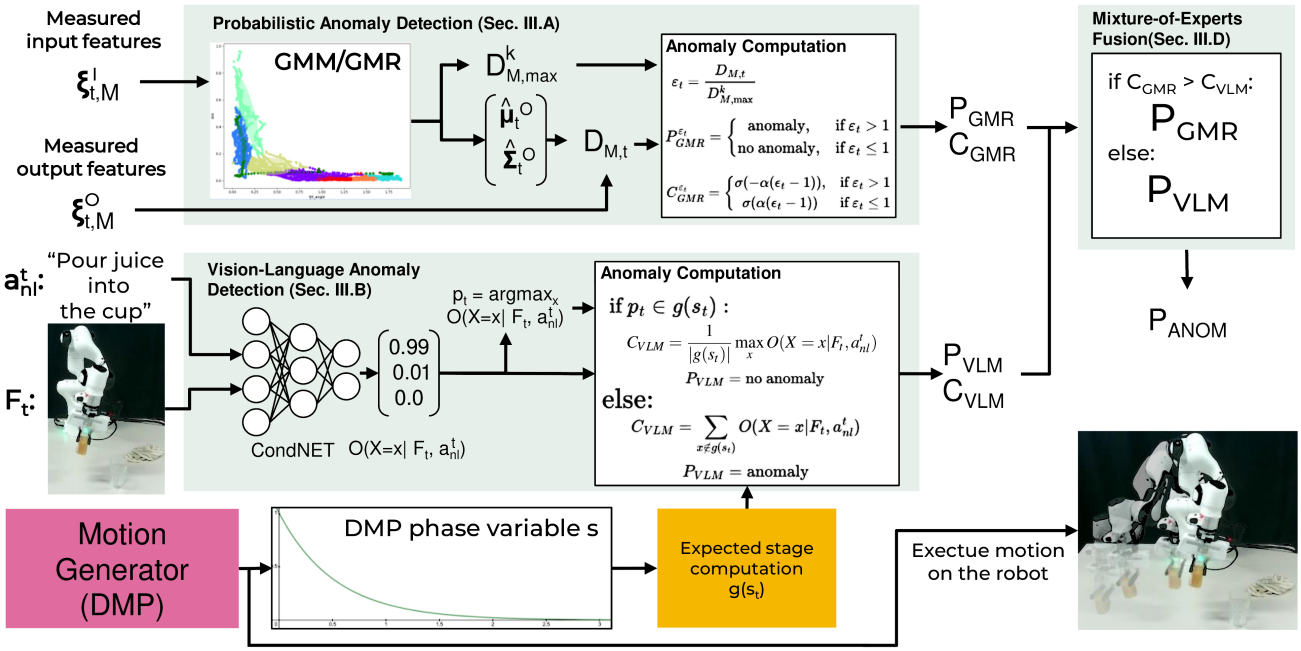


Fig. 2: **System Architecture.** After training the VLM and GMR experts, we developed a mixture-of-experts anomaly detection pipeline that combines their strengths. For the GMR expert, we use measured input features to regress the predicted feature value $\hat{\mu}_t^O$ and covariance $\hat{\Sigma}_t^O$. We compute the maximum Mahalanobis distance for the mixture component associated with the observed input features and the Mahalanobis distance between observed and predicted output features. These values are combined to compute the anomaly prediction and confidence score, where $\sigma(x)$ is the sigmoid function and α controls the confidence magnitude. For the VLM, we leverage knowledge of the current action and camera observations. The VLM regresses the current motion state, which we compare to the expected state. Unlike prior work [8], we express the expected state as a function of action progress, enabling continuous motion policies instead of segmented primitives. Finally, we implement a late fusion strategy, weighted by prediction confidence, to merge both experts' outputs.

distance to quantify the deviation from the expected output feature vector with

$$D_{M,t} = \sqrt{(\boldsymbol{\xi}^{\mathrm{O}}_{t,\mathrm{M}} - \hat{\boldsymbol{\mu}}^{\mathrm{O}}_t)^{\mathrm{T}} (\hat{\boldsymbol{\Sigma}}^{\mathrm{O}}_t)^{-1} (\boldsymbol{\xi}^{\mathrm{O}}_{t,\mathrm{M}} - \hat{\boldsymbol{\mu}}^{\mathrm{O}}_t)}. \tag{3}$$

Finally, to determine the anomaly prediction label P_{GMR}, we compute

$$\varepsilon_t = \frac{D_{M,t}}{D^k_{M,\max}}$$

by dividing the Mahalanobis distance of the observed feature vector $D_{M,t}$ by the highest observed Mahalanobis distance $D^k_{M,\max}$ for the training data assigned to mixture component k of the skill. We select k as the mixture component maximizing $\arg\max_k P(K=k|\boldsymbol{\xi}^{\mathrm{I}}_{t,\mathrm{M}})$ for the measured input feature vector. We set

$$P^{\varepsilon_t}_{GMR} = \begin{cases} \text{anomaly}, & \text{if } \varepsilon_t > 1 \\ \text{no anomaly}, & \text{if } \varepsilon_t \leq 1 \end{cases} \tag{4}$$

with prediction confidence

$$C^{\varepsilon_t}_{GMR} = \begin{cases} \sigma(-\alpha(\varepsilon_t - 1)), & \text{if } \varepsilon_t > 1 \\ \sigma(\alpha(\varepsilon_t - 1)) & \text{if } \varepsilon_t \leq 1 \end{cases}, \tag{5}$$

where $\sigma(x)$ is the sigmoid function $\frac{1}{1+e^x}$. Hyperparameter $\alpha \in \mathbb{R}_{\geq 0}$ can be chosen to scale the magnitude of the confidence score based on the number of observed demonstrations. We compute ε_t for all output sensor modalities and return $P_{GMR} = \textit{anomaly}$ if an anomaly is detected for at least one sensor modality. C_{GMR} is set to the maximum confidence score across all anomaly predictions or the maximum confidence score if all predictions returned *no anomaly* for all sensor modalities.

C. Vision-Language anomaly detection

Similarly, for each skill, we learn action preconditions and effects from the teleoperated dataset $\mathscr{D}_{tele}$, following [8]. The visual-language model takes as input the current frame F_t from an external camera and the natural language description of the skill, a^t_{nl}. Given (F_t, a^t_{nl}), it outputs a probability distribution $O(X = x \mid F_t, a^t_{nl})$ over three classes: {pre, effect, unsatisfied}. However, these predictions alone don't indicate anomalies—for example, predicting `pre` at the start of a motion is expected, while the same prediction at the end may signal an anomaly. In [8], this is addressed by segmenting each skill into primitive motions and assigning expected stages to each. But for complex or highly dynamic skills, such segmentation is difficult. Instead, we define the expected stage as a function of the motion phase.

The phase of a motion can be represented in various ways depending on the skill representation. For example, [27] uses path length. We use Dynamical Movement Primitives (DMPs) [28] to model motion policies, making the DMP phase variable s a natural choice.

DMPs describe each skill as a second-order mass-spring-damper system, modulated by a learned nonlinear forcing

function $f(s)$:

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f(s), \tag{6}$$

$$\tau \dot{y} = z, \tag{7}$$

$$\tau \dot{s} = \alpha_s s, \tag{8}$$

where y is the trajectory, z is an auxiliary variable, and s is the phase. The decay of s is controlled by α_s. We define the expected motion stage as a function of the DMP phase variable at time t, denoted s_t, i.e., $g(s_t)$, where $g : s_t \mapsto \{\text{pre}, \text{effect}, \text{unsatisfied}\}$. This mapping is manually specified per skill (e.g., if $0.8 < s_t < 1$, then $g(s_t) = \text{pre}$).

To determine whether an anomaly has occurred, we compare the current phase prediction, $p_i = \arg\max_x O(X = x|F_t, a^t_{nl})$, with the expected phase given by $g(s_t)$. If p_t belongs to the expected set, i.e., $p_t \in g(s_t)$, no anomaly is detected ($P_{VLM} = \text{no anomaly}$), and the anomaly prediction confidence is computed as

$$C_{VLM} = \frac{1}{|g(s_t)|} \max_x O(X = x|F_t, a^t_{nl}), \tag{9}$$

where $\frac{1}{|g(s_t)|}$ serves to scale the confidence score. During experiments, we observed frequent fluctuations in model predictions during motion transition phases, such as when pouring liquid into a cup. By introducing $\frac{1}{|g(s_t)|}$, we can further reduce confidence in these periods, making the system rely more on other experts. If the current model prediction does not match the expected set, i.e., $p_t \notin g(s_t)$, an anomaly is detected ($P_{VLM} = \text{anomaly}$), and the anomaly confidence is computed as the sum of the probabilities assigned to the anomalous classes:

$$C_{VLM} = \sum_{x \notin g(s_t)} O(X = x|F_t, a^t_{nl}). \tag{10}$$

D. Mixture-of-Experts

Finally, we use a late fusion strategy to combine the predictions of both models. Since our mixture-of-experts (MoE) model consists of only two experts, we adopt a simple winner-takes-all approach, where the expert with the higher confidence determines the final prediction. Specifically, if the GMM anomaly prediction confidence is higher than that of the VLM, we take the GMM prediction as the final decision. Conversely, if the VLM confidence is higher or equal to that of the GMM, we use the VLM prediction:

$$P = \begin{cases} P_{GMM} & \text{if } C_{GMM} > C_{VLM}, \\ P_{VLM} & \text{if } C_{VLM} \geq C_{GMM}. \end{cases} \tag{11}$$

IV. Experiments

A. Experimental setup

We develop two distinct experimental scenarios to evaluate the performance of the proposed mixture-of-experts anomaly detection framework across vastly different tasks. Specifically, we consider an industrial and a household scenario.

Our industrial scenario (Fig. 3) focuses on contact-rich manipulation, where the robot uses a specialized gripper to pick up a box. The task consists of four sequential skills:

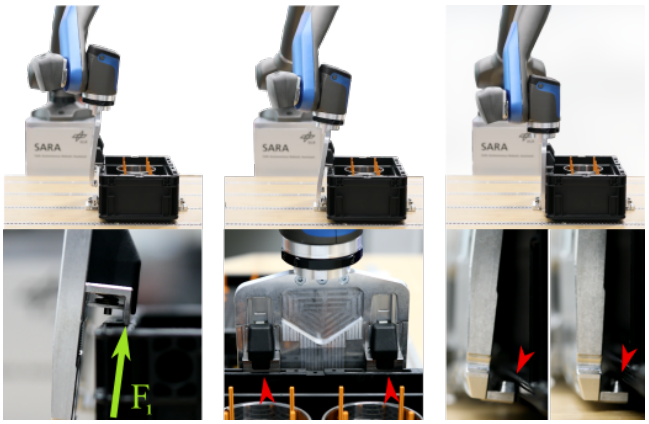
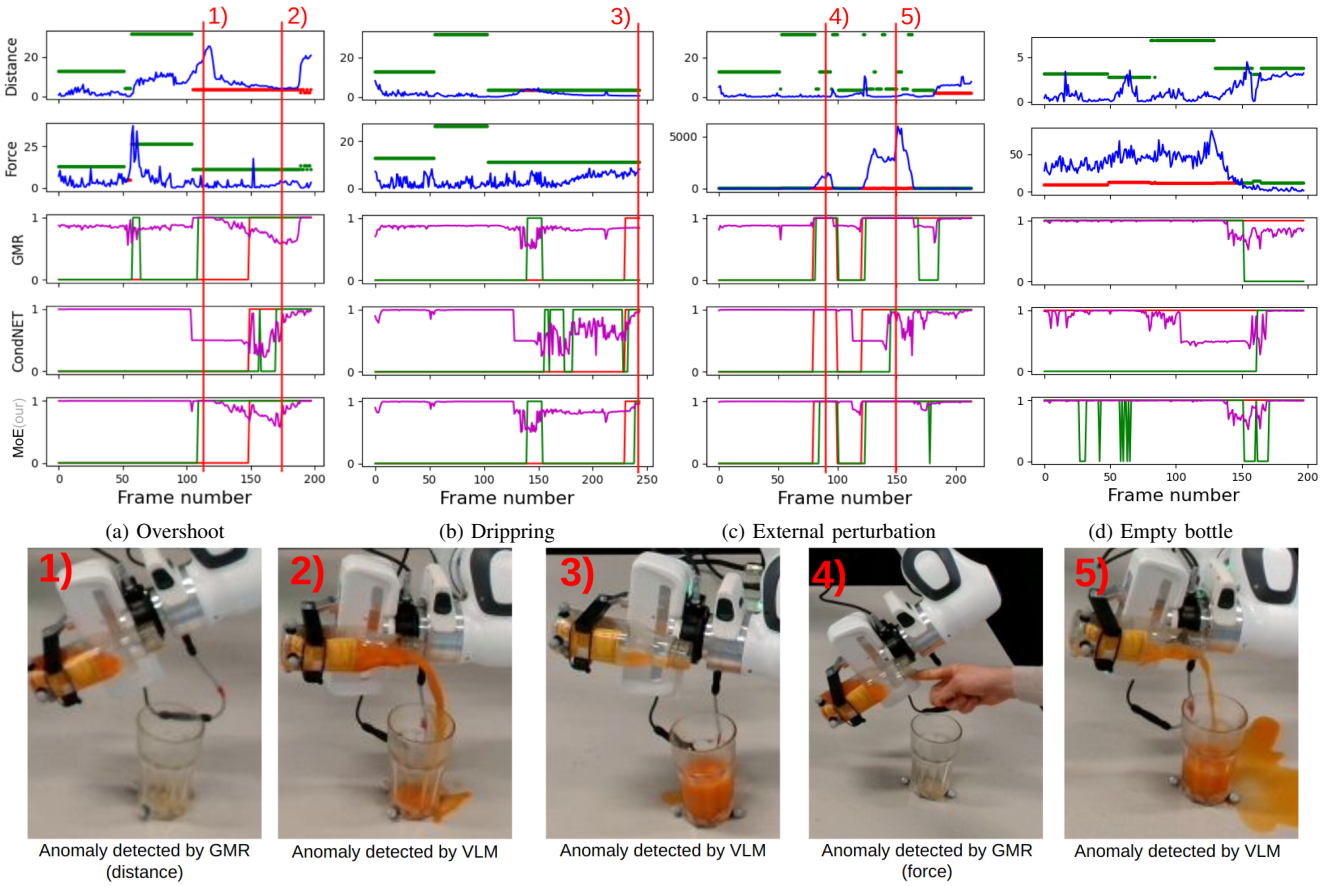


Fig. 3: The different phases and challenges of the box grasping and locking task. As shown in the left column, the robot must apply force via the front part of the slides to move them upward. It must maintain contact with the box's side wall until the middle configuration is reached, without getting too close, as this can block the sliding mechanism. Then, the locking pin must be pushed to compress the slide springs while rotating the gripper to a vertical position. Due to tight clearance between the box and the lower locks on the gripper, rotating too early can cause a collision.

(1) approaching the box, (2) pushing down along its side to tension the gripper's internal springs (left two columns in Fig. 3), (3) moving closer to lock onto the box, and (4) rotating into a vertical configuration to complete the grasp. The figure also outlines relevant details of each step. We define four anomaly types: (1) the gripper misses the box during approach, (2) the linear slides lock prematurely due to hardware failure, (3) a user pushes the gripper during free-space motion, and (4) a user pulls it during spring tensioning. ConditionNet is trained on 107 successful and 71 unsuccessful skill executions. The GMM-based method is trained on 3 kinesthetic and 3 autonomous executions (4 skills each), totaling 24 successful skill examples. For each skill, we encode the 6D end-effector pose and 3D contact force as a GMM with 2 components. During execution, the GMM is conditioned on the current pose to predict expected forces, which are compared to measured forces to detect anomalies. To evaluate detection performance, we collected an additional 47 successful and 35 unsuccessful skill executions across the defined anomaly types.

Our second scenario involves a juice-pouring task, where the robot's goal is to pour orange juice from a bottle into a glass. We consider two types of anomalies: (1) a spill occurs, and (2) the bottle is empty. Spills can result from various factors, such as the robot overshooting the cup, liquid dripping along the bottle's edge during the final phase of pouring, or external disturbances caused by a human. In this scenario, we train the ConditionNET model on the (Im)PerfectPour dataset [8], which contains 406 successful and 138 unsuccessful task demonstrations. Additionally, we collect 4 successful task executions using kinesthetic teaching to train the GMM-based anomaly detection model on the recorded end-effector pose relative to the pouring target and the force on the end-effector using 10 mixture components. Finally,



(a) Overshoot (b) Drippring (c) External perturbation (d) Empty bottle

(e) Snapshots from the executions.

Fig. 4: Anomaly prediction results for different scenarios in the pouring task. The first two rows display the Mahalanobis distance (blue) and the threshold values for the distance and force domains. An anomaly is triggered by the GMR-based detector when the threshold is exceeded. The next three rows show the prediction confidence (magenta) and anomaly detections from GMR, ConditionNet, and our MoE method (green). Ground truth anomalies are marked by a red line, where 0 indicates normal and 1 indicates an anomaly. Snapshots in Fig. 4e show the moments when GMR and VLM experts detect anomalies, with corresponding time steps marked by red vertical lines in the plots.

TABLE I: Evaluation of the frame-wise detection performance of our mixture-of-experts anomaly detection approach (MoE) against the baselines GMR-based [29], and ConditionNET [8] for the box-grasping and the pouring task.

	Box-grasping					
Method	Acc	Pre	Rec	F1	F1@50	Del
MoE (our)	88.1	96.6	**82.6**	**88.3**	**86.4**	**0.47**
GMR	**88.8**	**100**	81.7	87.4	78.9	1.20
CondNET	79.8	95.9	73.2	81.6	75.0	1.23
	Pouring					
Method	Acc	Pre	Rec	F1	F1@50	Del
MoE (our)	**88.7**	**88.7**	**88.1**	**87.2**	**84.7**	-0.3
GMR	84.5	86.9	81.0	83.3	76.7	**-0.4**
CondNET	75.8	88.0	67.2	70.2	69.3	0.4

we collect 27 failed autonomous executions to evaluate the performance of the anomaly detection models.

B. Quantitative Results

We evaluate the frame-wise anomaly detection performance of our proposed MoE detector against the individual anomaly detection approaches integrated within the framework. To assess the performance, we report frame-wise accuracy, precision, recall, F1 score, F1 at 50% overlap threshold, and detection delay in seconds averaged across all anomaly cases of a task. Detection delay is defined as the time interval between the first occurrence of an anomaly in the ground truth and the first time step at which the respective method triggers an anomaly. The F1 at 50 % overlap threshold considers an anomaly detected if the frame-wise intersection over union (IoU) of anomaly ground truth and detection is larger than 50%. To ensure confident anomaly classifications, we filter raw predictions of each method by taking the majority of predictions over a sliding time window of eight time steps.

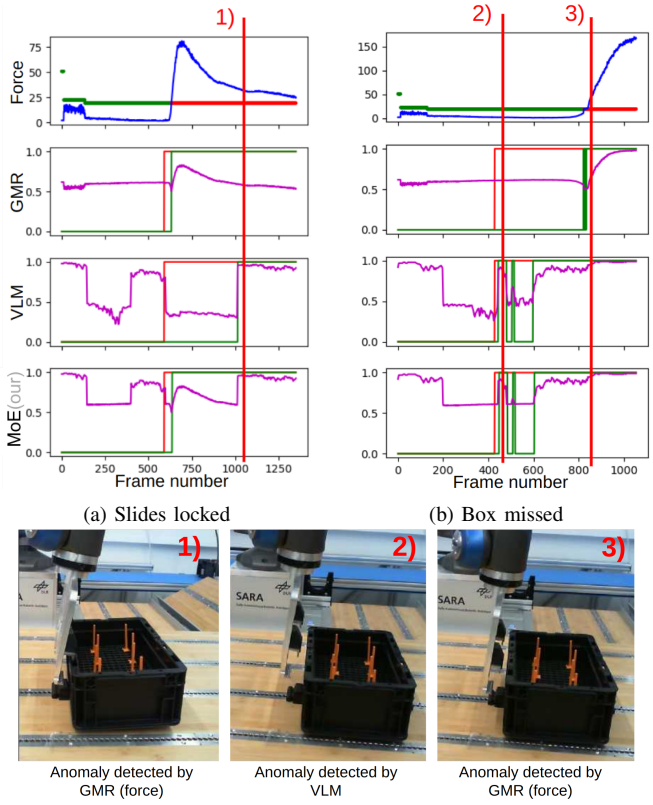
Table I presents the quantitative evaluation results across various anomaly cases caused by diverse sources for the box-grasping and pouring task. In the box-grasping experiment, the MoE approach achieves frame-wise detection performance comparable to the GMR-based method while reducing detection delay by more than 60% compared to both individual detectors. MoE also outperforms the other approaches

in the F1@50% score, indicating a more precise anomaly detection result. Notably, in the missed-box anomaly case, vision-based detection enables earlier anomaly identification compared to using only low-level features, whereas those are better suited to promptly detect force based anomalies (see Fig. 5).

For the pouring task, MoE outperforms both other approaches for all frame-wise detection scores while maintaining a detection delay similar to GMR. Both MoE and GMR report a negative detection delay, which highlights the early detection capabilities of these methods, warning the system before the effect of an anomaly becomes visually apparent. In this scenario, these approaches can detect an imminent liquid spill if the bottle in the gripper is tilted outside the expected pouring region of the cup, allowing the system to intervene proactively.

C. Qualitative Results

The early detection of a spill by our MoE detector can be observed in Fig. 4a, where the spill caused by overshooting the cup occurs at time step 150. However, since the robot leaves the expected pour region for the cup, while the bottle is already tilted (depicted in Fig. 4e-1)), the GMR-based detector in the upper row of Fig. 4a, monitoring the relative end-effector position, triggers an anomaly shortly after time step 100. The higher prediction confidence score of the GMR-based detector outweighs the ConditionNET result, leading to anomaly detection within the MoE framework. Conversely, in the first half of the task, ConditionNET has a higher confidence score, outvoting a false positive prediction of the GMR-based detector. The dripping liquid failure case in Fig. 4b occurs in the final pouring phase, as shown in Fig. 4e-3), when the bottle is nearly empty and the liquid sticks to the opening, flows along the bottleneck to then drip outside the cup. Since low-level sensor readings remain within the expected ranges, this anomaly is only visually detectable by ConditionNET, resulting in a true positive anomaly detection within the MoE framework. Around time step 150 during this task, a false positive anomaly detection is triggered, caused because both confidence scores of the experts are low and the measured distance at the beginning of the pouring crosses the boundary of the expected region. Fig. 4c and Fig. 4e-4) depict a scenario, where the robot is pushed away from the cup in two different phases of the task, which is reliably detected by the GMR's force-based monitoring approach shown in the second row. After the second perturbation, when the robot continues with the normal task execution, GMR classifies the situation as normal. However, since a spill has already occurred during the second push, ConditionNET correctly identifies the anomaly in Fig. 4e-5), leading to an overall correct classification by the MoE framework. The final anomaly case for the pouring task is shown in Fig. 4d, where the robot picks up an empty bottle. Since the expected force is not met during the first three quarters of the task, GMR continuously detects this deviation. Even though pouring with an empty bottle has not been present in the training data, ConditionNET detects



(a) Slides locked (b) Box missed

Anomaly detected by GMR (force) | Anomaly detected by VLM | Anomaly detected by GMR (force)

(c) Snapshots from executions.

Fig. 5: Anomaly prediction results for different scenarios in the industrial box grasping task are shown. The top row displays the computed Mahalanobis distance (blue) and the maximum Mahalanobis distance in the force domain. When the distance exceeds a threshold, the GMR-based detector flags an anomaly. The next three rows show the prediction confidence (magenta) and the anomaly detection results from GMR, ConditionNet, and our proposed Mixture of Experts (MoE) method (green). Ground truth anomalies are marked by a red line, where 0 indicates no anomaly and 1 indicates an anomaly. Additionally, Fig. 5c presents execution snapshots at the moments when GMR and VLM experts detect anomalies. These time steps are also marked with red vertical lines in the plots.

that the effect of the task is not met when no liquid is poured an triggers an anomaly in the final phase of the task, leading to an overall improved result with the MoE.

Two representative anomaly cases for the box grasping task are shown in Fig. 5. In Fig. 5a, a hardware defect prevents the gripper's slides from moving, resulting in an unexpected force signal that is correctly detected by the GMR-based method. Later, ConditionNet identifies the anomaly when the push-down skill fails to reach its effect phase in the final execution quarter (see Fig. 5c-1). The combined MoE approach significantly reduces detection delay compared to using ConditionNet alone. In the missed box case (Fig. 5b), ConditionNet detects the anomaly early, at the time step shown in Fig. 5c-2, when the gripper should have contacted the side wall. In contrast, the GMR-based method detects it later, only after the force signal confirms the missed contact (Fig. 5c-3). Again, the MoE approach shortens the detection

delay compared to GMR alone.

V. Conclusion

We demonstrated the need for an approach that fuses anomaly predictions from two experts with different detection mechanisms on both an industrial and a household task. The GMR-based approach reliably identified anomalies by analyzing deviating contact forces, whereas the VLM-based detector registered even subtle visual deviations in the scene. In the experiments, we showed that the GMR- and VLM-based detectors complement each other in different scenarios, leading with our proposed MoE to a reduction of the detection delay of up to 60% and an overall improved frame-wise detection performance. Future work could focus on integrating more modalities such as audio and depth perception, and combining further expert detectors with a modified voting system.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, July 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[2] C. Willibald, T. Eiband, and D. Lee, "Collaborative programming of conditional robot tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5402–5409.

[3] T. Eiband, C. Willibald, I. Tannert, B. Weber, and D. Lee, "Collaborative programming of robotic task decisions and recovery behaviors," *Autonomous Robots*, vol. 47, no. 2, pp. 229–247, 2023.

[4] T. Eiband, M. Saveriano, and D. Lee, "Intuitive programming of conditional tasks by demonstration of multiple solutions," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4483–4490, 2019.

[5] D. Romeres, D. K. Jha, W. Yerazunis, D. Nikovski, and H. A. Dau, "Anomaly detection for insertion tasks in robotic assembly using gaussian process models," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1017–1022.

[6] D. Park, H. Kim, and C. C. Kemp, "Multimodal anomaly detection for assistive robots," *Autonomous Robots*, vol. 43(3), pp. 611–629, 2019.

[7] D. Azzalini, A. Castellini, M. Luperto, A. Farinelli, and F. Amigoni, "Hmms for anomaly detection in autonomous robots," in *Int. Conf. on Autonomous Agents and MultiAgent Systems*, 2020, pp. 105–113.

[8] D. Sliwowski and D. Lee, "Conditionnet: Learning preconditions and effects for execution monitoring," *IEEE Robotics and Automation Letters*, 2024.

[9] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[10] Y. Yoo, C.-Y. Lee, and B.-T. Zhang, "Multimodal anomaly detection based on deep auto-encoder for object slip perception of mobile manipulation robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 443–11 449.

[11] A. Inceoglu, E. E. Aksoy, and S. Sariel, "Multimodal detection and classification of robot manipulation failures," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1396–1403, 2024.

[12] D. Altan and S. Sariel, "Clue-ai: A convolutional three-stream anomaly identification framework for robot manipulation," *IEEE Access*, vol. 11, pp. 48 347–48 357, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247476170

[13] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 36–51.

[14] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," in *Proceedings of The 2nd Conference on Lifelong Learning Agents*, ser. Proceedings of Machine Learning Research, S. Chandar, R. Pascanu, H. Sedghi, and D. Precup, Eds., vol. 232. PMLR, 22–25 Aug 2023, pp. 120–136. [Online]. Available: https://proceedings.mlr.press/v232/du23b.html

[15] A. Inceoglu, E. E. Aksoy, A. Cihan Ak, and S. Sariel, "Fino-net: A deep multimodal sensor fusion framework for manipulation failure detection," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6841–6847.

[16] X. Zhang, Y. Ding, S. Amiri, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, "Grounding classical task planners via vision-language models," in *arXiv preprint arXiv:2304.08587*, 04 2023.

[17] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone, "Real-Time Anomaly Detection and Reactive Planning with Large Language Models," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[18] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using gaussian mixture models," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007, pp. 1–8.

[19] C. Agia, R. Sinha, J. Yang, Z.-a. Cao, R. Antonova, M. Pavone, and J. Bohg, "Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress," *arXiv preprint arXiv:2410.04640*, 2024.

[20] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5.

[21] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds. Cham: Springer International Publishing, 2019, pp. 703–716.

[22] G. M. et al., "Pddl - the planning domain definition language," *Techincal Report*, 08 1998.

[23] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo, "AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=JVkdSi7Ekg

[24] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. Ambrus, H. Nishimura, and M. Itkina, "Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies," in *Proceedings of Robotics: Science and Systems*, Los Angeles, USA, June 2025.

[25] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2, pp. 115–131, Oct 2011. [Online]. Available: https://doi.org/10.1007/s10514-011-9234-3

[26] D. Sliwowski, S. Jadav, S. Stanovcic, J. Orbik, J. Heidersberger, and D. Lee, "Reassemble: A multimodal dataset for contact-rich robotic assembly and disassembly," *arXiv preprint arXiv:2502.05086*, 2025.

[27] J. De Schutter, "Invariant description of rigid body motion trajectories," *Journal of Mechanisms and Robotics*, vol. 2, no. 1, p. 011004, 11 2009. [Online]. Available: https://doi.org/10.1115/1.4000524

[28] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023. [Online]. Available: https://doi.org/10.1177/02783649231201196

[29] C. Willibald and D. Lee, "Multi-level task learning based on intention and constraint inference for autonomous robotic manipulation," 2022, pp. 7688–7695.