ILeSiA: Interactive Learning of Situational Awareness from Camera Input

Petr Vanc[1,*], Giovanni Franzese[2], Jan Kristof Behrens[1], Cosimo Della Santina[2], Karla Stepanova[1], and Jens Kober[2].

***Abstract*— Learning from demonstration is a promising way of teaching robots new skills. However, a central problem when executing acquired skills is to recognize risks and failures. This is essential since the demonstrations usually cover only a few mostly successful cases. Inevitable errors during execution require specific reactions that were not apparent in the demonstrations. In this paper, we focus on teaching the robot situational awareness from an initial skill demonstration via kinesthetic teaching and sparse labeling of autonomous skill executions as safe or risky. At runtime, our system, called ILeSiA, detects risks based on the perceived camera images by encoding the images into a low-dimensional latent space representation and training a classifier based on the encoding and the provided labels. In this way, ILeSiA boosts the confidence and safety with which robotic skills can be executed. Our experiments demonstrate that classifiers, trained with only a small amount of user-provided data, can successfully detect numerous risks. The system is flexible because the risk cases are defined by labeling data. This also means that labels can be added as soon as risks are identified by a human supervisor. We provide all code and data required to reproduce our experiments at imitrob.ciirc.cvut.cz/publications/ilesia.**

I. Introduction

Learning from demonstration has a huge potential to reduce the setup cost and increase the flexibility of robots thanks to the intuitive teaching of new manipulation skills. This involves guiding a robot kinesthetically through a task to acquire a *nominal* demonstration that the robot is expected to reproduce. A central problem remains to decide at every moment if it is safe to execute such a learned skill. Identification of situations that pose a risk requires a novel level of task-dependent situational awareness. Although there are works that focus on creating situational awareness for robots to enhance their planning capabilities [1], these typically do not consider the very fast feedback loops that are required in robotic skill execution. Timely identification and addressing of pending failures, we call them *risks*, can often prevent harm or lead to successful recovery.

In this paper, we address the problem of recognizing *risks* during skill execution from camera images given a very limited amount of supervision by the user. To this end, we propose a method called ILeSiA: Interactive Learning of Situational Awareness from Camera Input. With ILeSiA, the user supervises the first few trials of skill executions and annotates parts of the trial as *risky* or *safe*. Using these data, ILeSiA trains a Gaussian Process (GP) risk estimator which based on a low dimensional image representation predicts a level of for the respective timestep. After that, ILeSiA can be used to monitor the robot's skill executions in real-time, and the risk signal can be used to adapt the robot's behavior. In Fig. 1 we show an example of skill execution and ILeSiA risk predictions for manipulation task (opening door) involving the Robothon Challenge Box 2023[1]. We use the Robothon Box for experiments within this paper to evaluate the proposed ILeSia method.

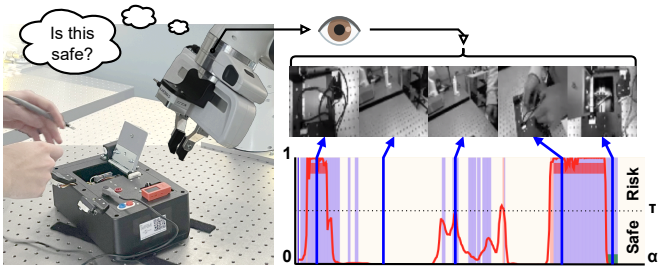

Fig. 1. An illustration showing the usage of the ILeSiA system. The robot assesses the potential risk of the situation based on the image. Manipulation tasks use the Robothon Challenge Box 2023 to demonstrate various skills. One example, 'Open a Door,' is illustrated, with a red overlay highlighting areas where risky behavior is likely.

The main contributions of this paper are:

- Development of a compact Risk Estimation Model: This model estimates the level of risk at any given timestep based on visual input from the camera, requiring only a single demonstration of the risk. It continuously assesses the environment for potential risks that could hinder the robot's task execution.
- Integration into a Learning from Demonstration (LfD) Framework: The risk awareness module has been implemented within an LfD framework, enabling the system to capture new demonstrations, retrain the model, and establish a feedback loop. This feedback loop allows the robot to adapt its behavior in real-time.

The code, an interactive visualisation tool together with other materials and results are available online at imitrob.ciirc.cvut.cz/publications/ilesia.

*The author conducted this research during his internship visit at TU Delft.

[1] are with Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Czech Republic {petr.vanc, jan.kristof.behrens, karla.stepanova}@cvut.cz

[2] are with Cognitive Robotics, Delft University of Technology, The Netherlands {G.Franzese, C.DellaSantina, J.Kober}@tudelft.nl

[1]Platonics: Robothon 2023 website: platonics-delft.github.io/

II. Related work

The field of fault detection in robotics is an active research area with many open questions. Numerous studies, like those by Van and Ge [2] and Wu et al. [3], emphasize the role of real-time processing in detecting mechanical or operational anomalies. Park et al. [4] employ a data-driven strategy to enhance anomaly detection in robotic manipulation, leveraging multiple sensory modalities. They train a hidden Markov model (HMM) on multimodal observations to identify deviations during varied manipulation tasks.

In contrast, our research centers on situational risks identified through visual inputs, expanding the scope to include not only operational faults (e.g., door failing to open when they should), but also situational awareness (e.g., human hands in the view). This involves the integration of perception, reasoning, and situational awareness, as suggested by Ruiz-Celada et al. [1], who advocate for enhanced robotic perception capabilities through smart integration of sensor data and reasoning technologies.

Further bridging the gap between traditional fault detection and modern risk assessment are interactive imitation learning (IIL) techniques, which provide a framework for robots to learn complex tasks through human feedback [5]. Techniques like DAgger (Data Aggregation) [6] and ThriftyDAgger [7] optimize the learning process by focusing on scenarios where human intervention is most critical, thereby reducing the expert burden as explored in FIRE (Failure Identification to Reduce Expert Burden) [8].

Distinctively, our approach also considers the automated handling of predicates and state estimation in contact-rich tasks [9], which contrasts with systems that rely on hand-crafted feature predicates. This enables a more dynamic and responsive system, capable of adjusting to new and unforeseen environmental conditions.

Overall, while traditional studies lay a solid foundation for understanding and detecting faults in robotic systems, our work extends these concepts into the broader and more complex domain of situational awareness through real-time video analysis, ensuring that robotic systems can detect risks and call the attention of a human using active learning or a human can intervene and correct situations that would not have been considered as safe before.

III. Method

A. Kinesthetic Demonstration

We assume that a trajectory for individual skill is recorded by kinesthetic teaching using the existing Learning from Demonstration (LfD) module[2], initially developed for the *2023 Robothon challenge* (refer to Fig. 1). The camera mounted on the end effector is recording a video during this demonstration. The LfD module enables the teaching and execution of robotic skills through kinesthetic demonstrations.

These newly acquired skills are stored and later performed as fixed trajectories within the task space coordinates. This ensures to have consistent camera views during the trajectory execution even when the target object moves between individual demonstrations. We call the originally recorded trajectory a *nominal* trajectory. The aim is to keep the model simple with short training and inference time so it is useful in industry and for Human-Robot Interaction, where the deviation from the designated path is considered risky. If the angle or distance relative to the target object would vary with each execution of the skill, the model would need to generalize across various views of the object, necessitating a more complex model.

[2]Platonics: Robothon 2023 website: platonics-delft.github.io/

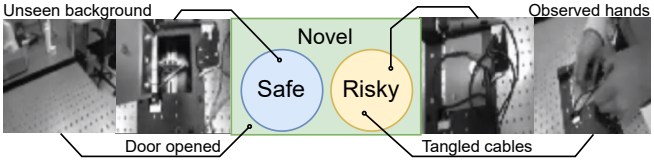

Fig. 2. Venn diagram example of variable space. Images are sorted as safe, risky, or novel.

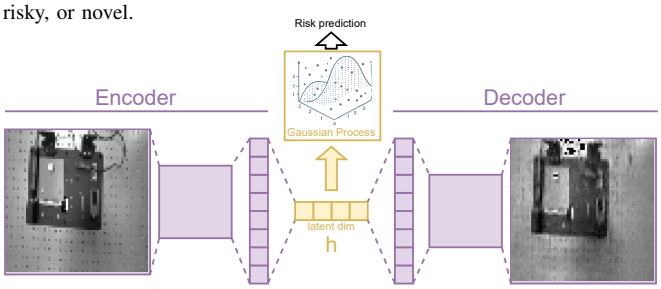

Fig. 3. Video embedding architecture utilizing a 4-layer autoencoder and its connection to the risk estimator.

B. Risky or Safe: Learning and Judging the current situation

The images observed during the demonstration can be classified into three categories: safe, risky, or novel (refer to the Venn diagram in Fig. 2). The goal of our model is to accurately classify incoming images into these categories based on training data, where images are explicitly labeled as either safe or risky.

To effectively detect risky situations during a novel demonstration, we initially process the video signal, resize it and convert to grayscale. Subsequently, we embed the input images into a latent space utilizing an *Autoencoder* network (see Fig. 3). This transformation not only reduces the data dimensionality but also enhances our model's ability to efficiently analyze and interpret the video data for potential risks.

Then we create an input vector $\mathbf{o}$ for the Risk Estimator by concatenating latent space vector $\mathbf{h}$ corresponding to the sample from the demonstration at time α, normalized time α, and Cosine distance d between the latent vector $\mathbf{h}$ and latent vector $\mathbf{h}^*$, corresponding to the sample from the nominal (kinesthetically learned) demonstration, at time α:

$$\mathbf{o} = \mathbf{h} \oplus \alpha \oplus d, \tag{1}$$

where $\oplus$ is concatenation. The real-time risk likelihood r for each skill is quantified using the following formula:

$$r = \mathcal{R}(\mathbf{o}), r \in [0, 1], \tag{2}$$

where $\mathcal{R}$ denotes the method used for risk estimation. We propose to use a Gaussian Process ($\mathcal{GP}$), which provides a probabilistic approach for capturing uncertainties and correlations within data. Especially its flexibility and robust interpolation capabilities might be enabling the risk estimator to detect various types of risks. GP is defined as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{3}$$

where $m(\mathbf{x})$ is the mean function specified as $m(\mathbf{x}) = 0$ and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function specified as Radial Base Function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_p^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\lambda^2}\right), \tag{4}$$

where σ_p^2 is the prior uncertainty of the model. The *length-scale* parameter λ determines how quickly the similarity (correlation) between inputs declines with distance.

The central challenge lies in the approach to handling novel situations. Novel situations are identified by detecting deviations from the distribution of the training samples, utilizing the model's inherent uncertainty. Images labeled either risky or safe are used for training.

When making posterior predictions (function $\mathbf{f}^*$) at new points $\mathbf{X}^*$ based on training data $\mathbf{X}$ and their ground truth risk labels $\mathbf{y}$, the Gaussian Process provides a posterior predictive distribution, which is also multivariate normal:

$$\mathbf{f}^* | \mathbf{X}, \mathbf{y}, \mathbf{X}^* \sim \mathcal{N}(\mu^*, \Sigma^*) \tag{5}$$

where:

$\mu^* = K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}$

$\Sigma^* = K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}K(\mathbf{X}, \mathbf{X}^*)$

σ_n^2 is the noise variance parameter, computed from all training samples (prior uncertainty). σ_p is set to 1 such that when you go out of distribution the sum of the two terms will also converge to one when going out of distribution. $K(\mathbf{X}, \mathbf{X})$ is the covariance matrix computed from the RBF kernel over all pairs of inputs in $\mathbf{X}$. I is an identity matrix.

Following the human approach to handling novel situations, we consider risky everything that is not safe, i.e., novel situations or situations corresponding to already labeled risky situations. To capture this by our model we propose a novel method for estimating risk r of the given situation:

$$r = \mu^* + \sigma^*, \tag{6}$$

where μ^* and σ^* are parameters of the predicted posterior distribution provided by the Gaussian process

Finally, the binary variable indicating the presence or absence of risk is computed given the selected threshold:

$$y = \begin{cases} 1 & \text{if } r > 0.5 \\ 0 & \text{if } r \leq 0.5 \end{cases}. \tag{7}$$

When the risk is detected (e.g., an unexpected object or human hand is detected), a robot is stopped and calls for attention of the teacher to handle the risky situation (see the following Sec. III-C).

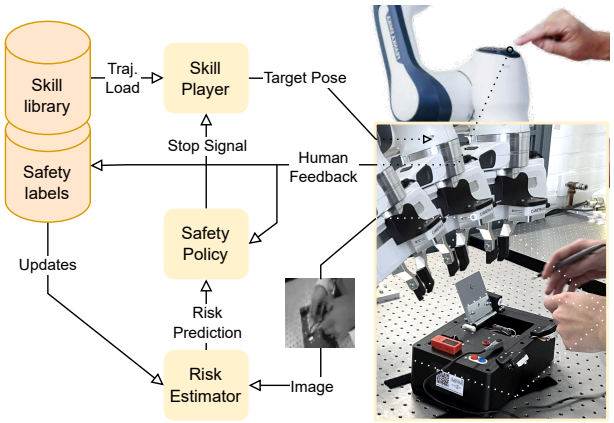

Fig. 4. Interactive learning loop.

C. Active and Interactive Labeling of situations from human feedback

The Risk Estimator is integrated into the Learning from Demonstration Module (see Fig. 4) as a Video Safety Module. The Risk Estimator is a key enhancement to the existing Learning from Demonstration (LfD) Module[2] (refer to Fig. 1).

The integration of the Risk Estimator into the Learning from Demonstration (LfD) package (as illustrated in Fig. 4) was designed to function autonomously, meaning each new skill execution provides fresh samples for learning the risk estimator. Any user interventions during a skill execution are automatically added to the training set including the provided labels. For instance, if an undesirable event occurs during a skill's demonstration and the user halts the execution, this interruption is attributed to the user.

Labeling is facilitated through interactive inputs either from the keyboard or directly using the Franka Emika robot's integrated buttons[3]. This method allows for immediate and accurate tagging of relevant data points as safe or risky during demonstrations.

IV. Experimental Setup

We test the proposed system in the context of manipulations of the Robothon Box (Fig. 1), which was part of the Robothon Challenge 2023 setup [10]. Our setup features a Franka Emika Panda robot with an Intel Realsense D400 series camera mounted on the end-effector. Robotic skills are taught via kinesthetic teaching and stored relative to the box. We execute the learned skills and record images with a frequency of $20Hz$ to ensure a comprehensive visual record.

The remainder of the Section describes the skills and failure modes (Sec. IV-A), and the video embedding (Sec. IV-B). This section is concluded with notes on the dataset collection for the risk estimator (Sec. IV-C)

[3]Franka Buttons package: github.com/franzesegiovanni/franka_buttons

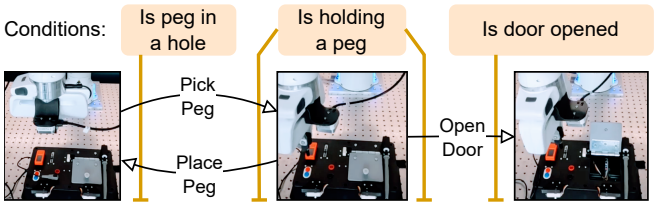

Fig. 5. Examples of recorded manipulation skills and their labeled conditions that are set to be predicted with risk estimator based on prepared dataset labels.

A. Recording manipulation skills

The experiments utilized specific manipulation tasks involving the Robothon Box, including: 1) Pick a Peg (see Fig. 5 (left)), 2) Open a Door (see Fig. 1), 3) Place a Peg, 4) Move Slider, and 5) Pick a probe. See our interactive tool[4] for examples of the tasks.

In the context of these tasks, we consider the following states or situations that depending on the task might pose a risk (see Figure 2 and complementary video material within the interactive tool provided on our website[4]):

1) Configuration of a door (opened/closed).
2) Incorrect rotation of a peg during the picking process.
3) Rotation of a peg after placement.
4) Configuration of a slider (start/end position).
5) Presence of obstacles or clutter, such as tangled cables, on the scene.
6) Visibility of human hands within the camera's view.

We group these situations into *known risks* (1 to 4), i.e., risks included and labeled in the training data, and *novel risks* (5 and 6), i.e., risks that the system hasn't been trained for.

For each task, we first recorded a demonstration via kinesthetic teaching and trained a robot skill from it. While recording the trajectories, we made sure that the camera was capturing enough information about the execution, such that a human watching the video stream could identify the risks. Then, we recorded four executions: one was safe, while the other three were manipulated to exhibit *known risks*. Additionally, we recorded four to five testing executions simulating real-world usage scenarios by additionally to the *known risks* also including the *novel risks* (e.g., tangled cables or human hands in the view).

In total, 5 kinesthetic demonstrations, 20 training executions (including various types of risks), and 22 testing executions were recorded[4].

B. Video embedding

We compare two different video embedding approaches: 1) the custom autoencoder described above and 2) a pre-trained ResNet-50 model [11].

For the autoencoder, the images from the camera are resized to 64x64 pixels and converted to grayscale to reduce computational complexity while preserving key visual details. Subsequently, the images are embedded into a latent space (as described in Sec. III-B).

The custom video embedding model (see Fig. 3) employs a 4-layer autoencoder architecture, comprising layers of convolution, normalization, *ReLU* activation, and max pooling operations. This embedding process compresses the input video signal into a compact latent space representation $H = (\mathbf{h})_{i=0}^{n}$ (n is a number of frames) with dimensions $\in \mathbb{R}^{n \times l}$, where l is latent space dimension.

The pre-trained ResNet-50 model [11] accepts RGB images with a resolution of 224x224 pixels and predicts the ImageNet class labels. To make a fair comparison, we upsample and normalize the grayscale image used for the autoencoder as RGB image. We extract as features the tensor values after either block 2 or 3, i.e., either 256 or 512, which are then used as embedding of the image and as input to the risk estimator.

C. Risk Estimator Dataset collection

The dataset

$$\mathcal{D} = \{d_i\}_{i=0}^{s} \tag{8}$$

for training the risk estimator is composed of s individual frames d_i (from all training executions), each represented as a triplet:

$$d_i = (\mathbf{h}_i, R_i, S_i) \tag{9}$$

where $\mathbf{h}_i$ is the feature vector extracted from the frame at index i, R_i is a binary risk label flag, and S_i is a binary safe label flag. Within each execution, there are two labeling windows (in the beginning and at the end) where the user labels the given situation as either risky or safe. Examples of these labeling windows for tasks 'Pick a peg' and 'Open door' are shown in Fig. 5.

Only labeled frames comprise the final dataset

$$\mathcal{D}_{\text{selected}} = \{(d_i \in \mathcal{D} \mid (R_i \vee S_i) = 1\}. \tag{10}$$

Additionally, all frames recorded during the kinesthetic demonstration are considered as *safe*. Unlabelled samples are discarded to not make unreasonable assumptions about the labeling process.

V. Experiments

In this section, we present the experiments conducted to validate the proposed method and individual design decisions. The experiments evaluate two different embedding techniques and two different risk estimation methods. We use the datasets described in Sec. IV, i.e., five tasks with each train and test executions including the described risks (see Sec. IV-A).

The following experiments are presented: Sec. V-A presents the hyperparameter selection for the architecture and the training procedures. In Sec. V-C, we present results regarding the impact of embedding quality. Then, Sec. V-B analyses the performance of the proposed method and the baseline on *known risks*. Sec. V-D continues with *novel risks*.

[4]Interactive tool for visualization of recorded trajectories and estimated risk: imitrob.ciirc.cvut.cz/publications/ilesia/video

A. Experiment configuration

In this section, we present our choices for hyperparameters for the experiments, i.e., the baseline architecture, the latentspace dimension, and the training parameters.

1) Risk estimation baseline: We compared the proposed method based on Gaussian processes (as described in Sec. III-B, Eq. (6)) with a multi-layer perceptron (MLP) as baseline. The MLP architecture includes 2 hidden layers, where each hidden layer has 30 nodes. Binary cross entropy loss is used with Adam optimizer and learning rate $1e-4$.

2) Size of Latent Space $\mathbf{h}$*:* The size of the latent space of the autoencoder has significant impact on the overall performance of the proposed method. To enable reasonable embedding quality (as seen by the reconstruction images) the latent space needs sufficient size to store the state and viewpoint and appearance of the input images. On the other hand, the performance and training times of GPs degrade with increasing size of the input dimensions. Informed by initial experiments, we settled to use a 32-dimensional latent space.

3) Number of training epochs: The number of training epochs is dynamically chosen based on the validation loss on an independent demonstration trial. We train typically for more than 2000 epochs.

B. Performance for the known risks

First, we evaluated the ability of the model to detect in testing trajectory the known risks, i.e., risks that were involved in the training datasets. Experiments compare the performance of Gaussian Process (GP) and Multi-Layer Perceptron (MLP) models in terms of their ability to detect risky situations accurately. In Fig. 6 we compare GP and MLP risk estimation models for one risky and one safe part of the testing demonstration. We show the predictions of the risk estimators along with human labels. Notably, the Gaussian process can provide stable risk estimation during the labeled window and the risk levels are also reasonable outside this human-labeled window. For example, the door was open even before the human labeled it as safe, so it is reasonable that the model predicts this area as safe. We can observe spikes in the predictions by the MLP method corresponding to wrong risk estimates.

When evaluating all the manipulation skills considered (see Sec. IV-A), we observed that GP model was consistently performing well on the testing demonstrations (over 96% accuracy on the samples from the labeled area). The observed errors were due to poor video reconstruction (see the Sec. V-C). MLP performed better in detecting smaller image features important for the risk assessment - e.g., it was able to correctly identify whether the peg was held or not in the gripper even in poorly reconstructed scenarios (i.e., those where a human has trouble understanding the reconstruction).

C. Impact of Reconstruction Quality

The quality of video reconstruction critically influences the ability of models to detect risky behaviors. Poor reconstruction quality often leads to misclassifications, as demonstrated by Fig. 7 (right). You can see that the GP can handle the poor reconstruction better than the MLP model. This highlights the importance of robust feature extraction in training data. Testing latent space sizes ranging from 8 to 48 revealed that sizes 16 and above could reconstruct risky behaviors in sequences up to 700 frames. However, smaller dimensions, such as size 12, failed to accurately reconstruct details like the peg's cable or slider position changes. Larger dimensions, such as 48 or 64, showed some success in capturing various peg rotations but struggled with comprehensive generalization across an infinite range of peg angles. In scenarios with a limited number of states, like a door being open or closed, reconstructions were significantly more accurate.

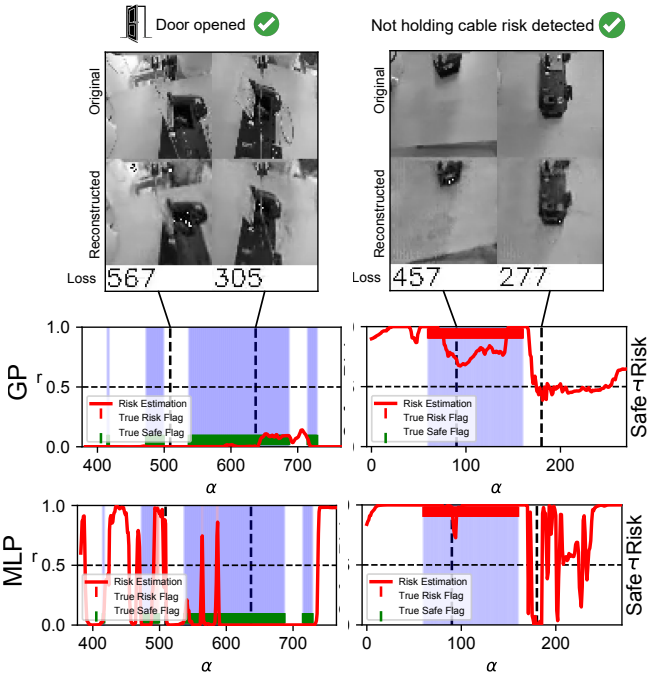

Fig. 6. Prediction on known risks on two test trials. Risk estimation based on Gaussian processes (GP) is compared to multilayer perceptron (MLP). Human labels are visualised (Safe area - green bar, Risky area - red bar). Accurate risk estimation is depicted by blue background and error by red background. Original and reconstructed images are shown together with the reconstruction loss value. Left: The door is opened as expected. GP does correctly predict it as safe. Right: The missing peg in the gripper correctly triggers risk. We observe spikes in the risk estimation for the MLP method outside the trained area as well as within the labeled one, corresponding to the wrong predictions. GP reasonably predicts also the areas outside the labeled area.

a) ResNet Comparison: : A comparison with the pre-trained ResNet-50 model revealed that it performs comparably to the Autoencoder-based video embedding, particularly when utilizing features extracted at the end of stages 2 and 3. However, the effectiveness of these features varies with the complexity of the risk behavior being analyzed. For instance, detecting the holding cable typically requires features from earlier stages, which capture more primitive shapes. Conversely, distinguishing between an open or closed door demands features from later stages that encapsulate

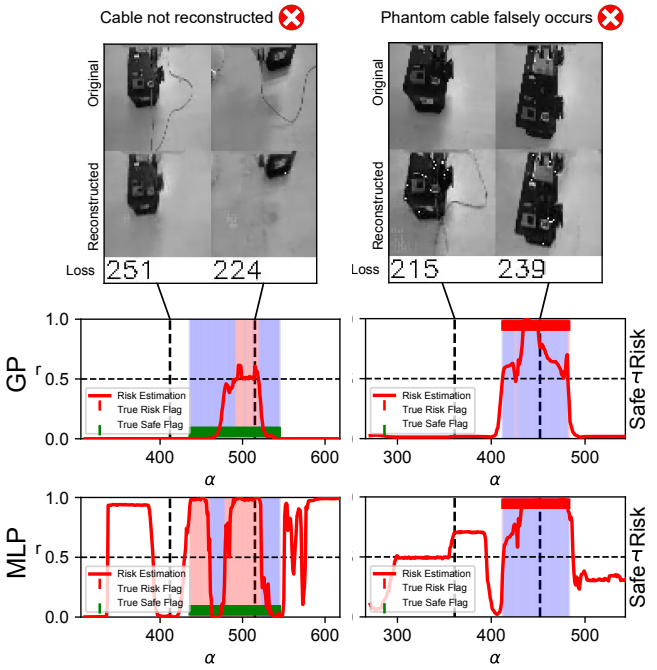

Fig. 7. Degradation of the performance due to incorrect video reconstructions. Risk estimation based on Guassian processes (GP) is compared to multilayer perceptron (MLP). Human labels are visualised (Safe area - green bar, Risky area - red bar). Accurate risk estimation is depicted by blue background and error by red background. Original and reconstructed images are shown together with the reconstruction loss value. Left: The cable in the original image is not reconstructed, causing the risk trigger. Right: Phantom cable that is not observed in the video is encoded into a latent vector and is triggering risk detection for the MLP method.

more complex visual information.

D. Quality of the risk estimation for novel risks

In this experiment, we evaluate the ability of the proposed model to handle novel risks, i.e., risks that were not present during the training demonstrations. An example of such a risk might be a novel object appearing in the scene (e.g., human hands, tangled cables, etc.) as well as different state of the object then we would expect during the demonstration (e.g., open door where it should be closed or wrongly rotated peg so it cannot be grasped by the gripper). Our model based on Gaussian processes (see Sec. III-B, Eq. (6)) is designed so that it can handle such a situation, and any deviation from a *nominal* trajectory is classified as risky. We can set also a specific threshold that determines the magnitude of change from the distribution required in an image to trigger a risk alert. This threshold can be adjusted with length scale parameter λ. Adjusting this setting might vary based on the desired behavior and required safety measures during execution. For example, if we set the parameter to 0, we will consider all novel situations as safe (i.e., optimistic model) on the other hand setting this parameter to high values might meet high safety standards, however would also lead to excessive numbers of false positive detections if the environment undergoes frequent changes (e.g., due to illumination) or if the reconstructions of the model are not perfect. We use as default value of 1 for this parameter.

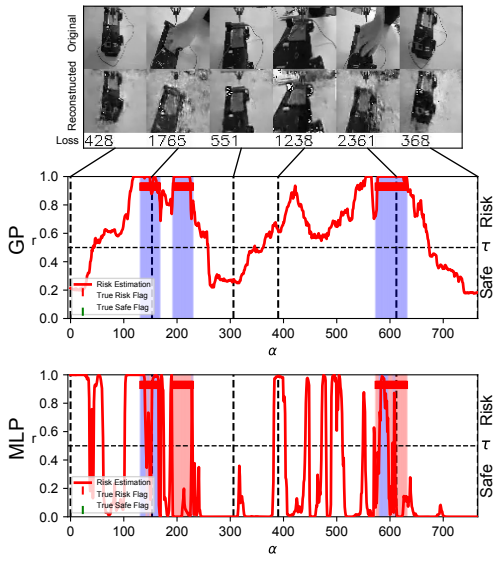

Fig. 8. Performance on novel risks. Comparison of Gaussian process (GP) model (top) and MLP model (bottom). Human labels are visualised (Safe area - green bar, Risky area - red bar). Accurate risk estimation is depicted by blue background and error by red background. Original and reconstructed images are shown together with the reconstruction loss value.

Fig. 8 illustrates the performance of the proposed model on the trajectory including novel risk (hands in the scene). The top part of the figure displays an inherent uncertainty detected in the GP model, where the hands appearing are causing an increase in the intrinsic uncertainty of the model. The bottom part shows the performance of the MLP model which cannot detect these risks accurately. The GP model can effectively detect novel risks when employing smaller latent dimensions. Using latent dimensions larger than size 32 does not facilitate sparse connections between sample points, leading to rapidly decreasing length scales and consequently, high uncertainty in similar images.

VI. Conclusion

In this paper, we have introduced a method to detect risky situations during robotic manipulations. We demonstrate our method in a learning-from-demonstration setting, where a robot is taught to manipulate a Robothon Box. The human teacher subsequently supervises a few executions of the learned skill and provides labels for encountered situations. Our method is based on Gaussian Processes and can detect the labeled and also novel risks in future manipulations. This paper widens the range of tasks where robots can safely and confidently act autonomously. In the future, we will make our method more robust by incorporating more signals into the risk estimation. For example, the reconstruction error of the autoencoder can yield valuable insights. Also, the tuning of the hyperparameters could be simplified. Finally, we are also interested in incorporating the proposed method into the online deliberation functions of a robot to take advantage of the increased situational awareness and gather more experiences with the system.

REFERENCES

[1] O. Ruiz-Celada, A. Dalmases, I. Zaplana, and J. Rosell, "Smart perception for situation awareness in robotic manipulation tasks," *IEEE Access*, vol. 12, p. 53974–53985, 2024.

[2] M. Van, S. S. Ge, and H. Ren, "Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, p. 1681–1693, Jul. 2017.

[3] L. Wu, W. Luo, Y. Zeng, F. Li, and Z. Zheng, "Fault detection for underactuated manipulators modeled by markovian jump systems," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, p. 4387–4399, Jul. 2016.

[4] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 407–414.

[5] C. Celemin, R. Pérez-Dattari, E. Chisari, G. Franzese, L. de Souza Rosa, R. Prakash, Z. Ajanović, M. Ferraz, A. Valada, and J. Kober, "Interactive imitation learning in robotics: A survey," *Found. Trends Robot*, vol. 10, no. 1–2, p. 1–197, nov 2022. [Online]. Available: https://doi.org/10.1561/2300000072

[6] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[7] R. Hoque, A. Balakrishna, E. R. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," in *Conference on Robot Learning*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237371142

[8] T. Ablett, F. Mari'c, and J. Kelly, "Fighting failures with fire: Failure identification to reduce expert burden in intervention-based learning," *ArXiv*, vol. abs/2007.00245, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220280267

[9] T. Migimatsu, W. Lian, J. Bohg, and S. Schaal, "Symbolic state estimation with predicates for contact-rich manipulation tasks," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1702–1709, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247244944

[10] P. So, A. Sarabakha, F. Wu, U. Culha, F. J. Abu-Dakka, and S. Haddadin, "Digital robot judge: Building a task-centric performance database of real-world manipulation with electronic task boards," *IEEE Robotics & Automation Magazine*, 2024.

[11] B. Koonce, *ResNet 50*. Berkeley, CA: Apress, 2021, pp. 63–72. [Online]. Available: https://doi.org/10.1007/978-1-4842-6168-2_6