

# A Multimodal Handover Failure Detection Dataset and Baselines

Santosh Thoduka<sup>1,3</sup> and Nico Hochgeschwender<sup>2</sup> and Juergen Gall<sup>3,4</sup> and Paul G. Plöger<sup>1</sup>

**Abstract**—An object handover between a robot and a human is a coordinated action which is prone to failure for reasons such as miscommunication, incorrect actions and unexpected object properties. Existing works on handover failure detection and prevention focus on preventing failures due to object slip or external disturbances. However, there is a lack of datasets and evaluation methods that consider *unpreventable* failures caused by the human participant. To address this deficit, we present the multimodal Handover Failure Detection dataset, which consists of failures induced by the human participant, such as ignoring the robot or not releasing the object. We also present two baseline methods for handover failure detection: (i) a video classification method using 3D CNNs and (ii) a temporal action segmentation approach which jointly classifies the human action, robot action and overall outcome of the action. The results show that video is an important modality, but using force-torque data and gripper position help improve failure detection and action segmentation accuracy.

## I. INTRODUCTION

An object handover between a human and a robot is a common action during tasks involving human-robot collaboration and physical human-robot interaction. It requires both the giver and the receiver to communicate and coordinate their actions to ensure a safe and successful handover. Failures in object handovers are likely, even for human-human handovers, for example, due to miscommunication, incorrect interpretation of intentions, object properties, or improper behaviour of either participant. Detecting failures enables robots to react appropriately - either by attempting to recover from the failure or informing the person about the failure. Certain types of failures, such as dropped objects, might cause harm, and thus detecting and handling them is important from a safety perspective.

Current approaches to error handling and failure detection in object handovers focus on the *approach* and *transfer* phase of the interaction, such as detecting slipping objects [1], increasing robustness to unwanted disturbances [2] and adapting to perception uncertainties [3]. Other approaches include using tactile and force-sensing to reliably release the object when pulled [4], reactive handovers for unknown objects [5], adapting to the human’s motion during the handover [6],

This work is supported by the European Union’s Horizon 2020 research and innovation program under grant agreement No. 871252 (METRICS), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GA 1927/4-2 (FOR 2535), the ERC Consolidator Grant FORHUE (101044724), and the Graduate Institute at Hochschule Bonn-Rhein-Sieg.

<sup>1</sup>Hochschule Bonn-Rhein-Sieg, Germany {santosh.thoduka, paul.ploeger}@h-brs.de

<sup>2</sup>University of Bremen, Germany nico.hochgeschwender@uni-bremen.de

<sup>3</sup>University of Bonn, Germany gall@iai.uni-bonn.de

<sup>4</sup>Lamarr Institute for Machine Learning and Artificial Intelligence, Germany



Fig. 1. Examples of failed robot-to-human object handovers due to a dropped object (top) and a person ignoring the robot (bottom).

and estimating the properties of the container being received and appropriate grasp locations [7]. While these approaches are concerned with improving the reliability of handovers and preventing failures, in some cases, failures are not preventable, particularly when they are caused by actions of the human participants. Therefore, failure *detection* should be considered in conjunction with failure *prevention* strategies. Introducing failure conditions is an important aspect of benchmarking robots and evaluating their performance in realistic conditions. Similar to software testing, subjecting a robot to abnormal conditions and injecting faults during a benchmark can be used to build arguments regarding its reliability and fault tolerance. In the METRICS HEART-MET competition for healthcare robots<sup>1</sup>, benchmarks are developed to specifically include failure modes as independent variables in the benchmarking protocol [8]. Failures are of particular relevance in healthcare contexts since the human partner might have cognitive or physical impairments, leading to failures such as dropped objects, as shown in Fig 1.

Datasets that include failures in robotics are quite rare, partly because it is difficult to collect such datasets. Failures typically have to be induced in order to balance the normal and failed instances since natural failures might occur rarely, and some types of failures might damage the robot or objects, or cause harm to people.

In this paper, we introduce the Handover Failure Detection (HFD) dataset in order to address the requirement for failure detection benchmarks for object handovers. The types of handover failures in the dataset formed part of the METRICS HEART-MET competition at ICRA 2023 as possible behaviours of persons for whom a robot had to fetch an object. The dataset has also been used to create a handover failure detection benchmark hosted on Codabench [9]. We focus on failures that are caused by the human participant in both robot-to-human (R2H) and human-to-robot (H2R)

<sup>1</sup><https://metricsproject.eu/healthcare>

TABLE I  
OBJECTIVES OF THE GIVER AND RECEIVER DURING R2H AND H2R HANDOVERS AND THE ASSOCIATED FAILURE MODES

	Phase	Objective	Failure modes
R2H	Approach	the giver approaches the receiver with the object the receiver approaches the giver	the human does not approach the robot
	Transfer	the receiver grasps the object the giver releases the object once it is safe to do so	the human does not grasp the object the object is dropped
	Retract	the giver releases the object and moves away from the receiver the receiver continues to grasp the object	the object is dropped
H2R	Approach	the giver approaches the receiver with the object the receiver approaches the giver	the human does not approach the robot the object is dropped
	Transfer	the receiver grasps the object the giver releases the object once it is safe to do so	the object is dropped
	Retract	the giver releases the object and moves away from the receiver the receiver continues to grasp the object	the human does not release the object

handovers. The dataset includes multimodal data such as video, robot joint state, and readings from a force-torque sensor. Multimodal data is important for handovers since both visual cues and physical signals (e.g. torque) influence the coordination [10]. We also present two baseline approaches for detecting the failures: the first uses video classification with 3D CNNs and the second uses temporal action segmentation to jointly classify the actions of the human participant, actions of the robot, and the overall outcome of the trial, both of which show the importance of learning from multimodal data.

The contributions of this paper are: 1) the multimodal HFD dataset consisting of human-induced failures during R2H and H2R handovers and 2) baseline methods for detecting handover failures and recognizing the type of failure using video classification and action segmentation methods<sup>2</sup>.

## II. RELATED WORK

The survey on object handovers by Ortenzi et al. [11] identifies works which consider error handling during handovers; most approaches that consider failures in handovers typically focus on the physical handover phase, which is concerned with a failure to coordinate between the receiver and the giver. For example, both Eguluz et al. [4] and Davari et al. [2] try to differentiate between an object being pulled (as part of the nominal R2H task) versus other disturbances, so that the robot does not release the object unexpectedly. Similarly, Parastegari et al. [12], [1] detect slipping objects and regrasp them to prevent failures caused by objects falling. Rosenberger et al. [13] prevent failures by aborting the handover if certain error conditions are met (such as the human moving too close to the grasp position, or if a collision is detected). Overall, the survey reports that only a minority of approaches incorporate error handling in their work, with the primary focus being to detect disturbances and preventing falling objects.

Liu et al. [3] describe approaches to increase safety by adapting the behaviour of the robot depending on the uncer-

tainty of its perception systems. For example, to prevent collisions in darker environments, the robot maintains a larger distance to the human since the uncertainty of their skeleton detection system increases. Han and Yanco [14] consider unrecoverable handover failures in their work, but investigate the expectations that a person has when a failure occurs. They find that people expect the robot to provide explanations, and not just non-verbal cues, for why a failure has occurred. In other work [15], [16], specific failures, such as occlusions or no detections of the human’s hands, are considered to either disregard an experimental trial, or to abort the handover. Iori et al. [17] consider two scenarios which deviate from a nominal handover, in which (i) the person pauses while approaching the robot before continuing, and (ii) the person’s arm is pushed by another person during the approach. The paper presents a reactive trajectory generation method which is able to adapt to the perturbations. Grigore et al. [10] aim to increase safety in R2H by considering human engagement signals (e.g. gaze) in addition to physical signals (e.g. current and torque) to decide if and when to release the object, and Mohandes et al. [18] use vision and joint torques to detect the person’s intentions and actions.

Prior research related to handover failures are concerned with preventing failures, or detecting conditions which might lead to a failure, and typically only deal with failures in the pre-handover and the physical handover phases. The post-handover phase should also be considered as it might take some time for the receiver to stabilize the object in their hand and get accustomed to its physical properties such as temperature or weight. In our work, we are concerned with detecting failures after they have occurred, with a specific emphasis on human-induced failures. We also consider failures in the post-handover phase for both H2R and R2H.

## III. THE HANDOVER FAILURE DETECTION DATASET

In this section, we introduce the HFD dataset, which consists of successful and failed R2H and H2R object handovers. The primary motivation for creating the dataset is to improve the monitoring capabilities of robots, thus enabling them to react to failure conditions appropriately.

<sup>2</sup>The dataset, code and Codabench benchmark are available at: [https://sthoduka.github.io/handover\\_failure\\_detection](https://sthoduka.github.io/handover_failure_detection)

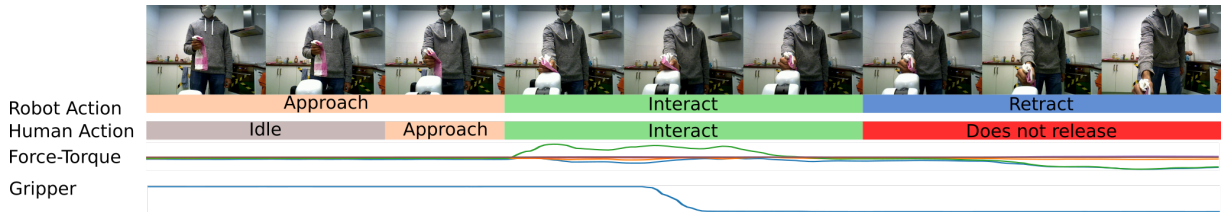


Fig. 2. The dataset consists of video, force-torque readings and robot joint states, and contains annotations for the robot and human actions.

### A. Development

The failures in the dataset are induced by the participants based on prior instructions. We select the failures by following the procedure defined in [8] for benchmarking robots by inducing failure scenarios. The procedure requires defining the objective of the task and identifying failure modes, which may be associated with negations of the objective. The objective of a handover task is to “safely transfer an object from the giver to the receiver”; this is further decomposed into the objectives of the three phases of a handover, *approach*, *transfer*, *retract*, as shown in Table I. The failure modes in Table I are identified by considering the objective of the human participant and specifying conditions under which the objective is not satisfied. Other independent variables, which are varied for each trial, include the robot platform, person, person’s pose, object, and location.

We use real robots and human participants instead of simulation to provide a more realistic benchmark. In the future, simulated data can be used as additional training data. The HandoverSim framework [19] is a promising approach in this direction, where the simulation of the human’s hand is derived from a motion capture dataset. If this is extended to simulate an entire human realistically, it would be possible to create large-scale training data in simulation where failures can be induced in a controlled manner. Nevertheless, it is important to evaluate on real robots, which is the purpose of the HFD dataset.

### B. Participant Recruitment

We recruited a total of 17 participants, all of whom were students familiar with robots. Participation was voluntary with no monetary or other compensation. Informed consent for the use of their data was obtained from the participants before starting the data collection.

### C. Procedure

For a given trial, the procedure is as follows: (i) the participant is instructed before the start of the trial about their desired behaviour, which may or may not induce a failure; (ii) data recording is started and the robot executes the handover action; (iii) the robot releases or grasps the object if a threshold is reached on the force-torque sensor; (iv) the robot retracts its arm and data recording is stopped.

### D. Dataset Characteristics

The dataset consists of 589 trials performed with two robots, 17 participants and 22 object classes. The class distribution and distribution between handover type and robot

TABLE II  
DATASET STATISTICS

	R2H					H2R				
	success	no approach	no grasp	drop	total	success	no approach	no release	drop	total
Toyota HSR	68	50	49	58	225	51	46	57	66	220
Kinova Gen3	18	17	17	20	72	19	18	17	18	72
<b>Total</b>	<b>86</b>	<b>67</b>	<b>68</b>	<b>76</b>	<b>297</b>	<b>70</b>	<b>64</b>	<b>74</b>	<b>84</b>	<b>292</b>

platform is summarized in Table II. The objects used include household objects such as a book, towel, sponge, bowl, bottle, etc.; a full list can be found in the datasheet which is included with the dataset.

The data for each trial includes RGB video from a robot-mounted camera, joint states of the robot arm (position, velocity and effort for each joint, including the gripper) and force-torque (F-T) sensor data<sup>3</sup>. In addition to the raw sensor data, resampled joint states and F-T data which are aligned to the timestamps of the video frames are included.

The training, validation and test sets are split such that a participant is present in only one of the sets, resulting in 337 trials for training, 101 for validation, and 151 for testing. In the approaches presented in the next section, all results are reported on the test set, with the validation set being used for hyperparameter selection and for early stopping.

### E. Annotations

The robot’s actions (idle, approach, transfer, retract, post-idle) are extracted based on the robot’s joint states<sup>4</sup>. The person’s actions are manually annotated, including the handover phases and non-nominal actions (idle, approach, transfer, retract, post-idle, not released, dropped). The start and end of each action is annotated, such that each frame of the video is associated with a robot action and a human action (see example in Fig. 2). The transfer phase was annotated by observing the F-T data and video to determine the start and end of the physical interaction of the person with the object and robot. The start of the approach phase and the end of the retract phase were annotated by observing the motion of the

<sup>3</sup>A wrist-mounted force-torque sensor on the Toyota HSR, and simulated force-torque readings at the wrist derived from the individual torques of the joints of the Kinova arm

<sup>4</sup>The robot actions always follow the same temporal order, therefore only the boundaries between the actions are determined based on whether the arm is moving or not. As such, the actions of the robot are assumed to be known and can also be retrieved from an action executor, such as a state machine.

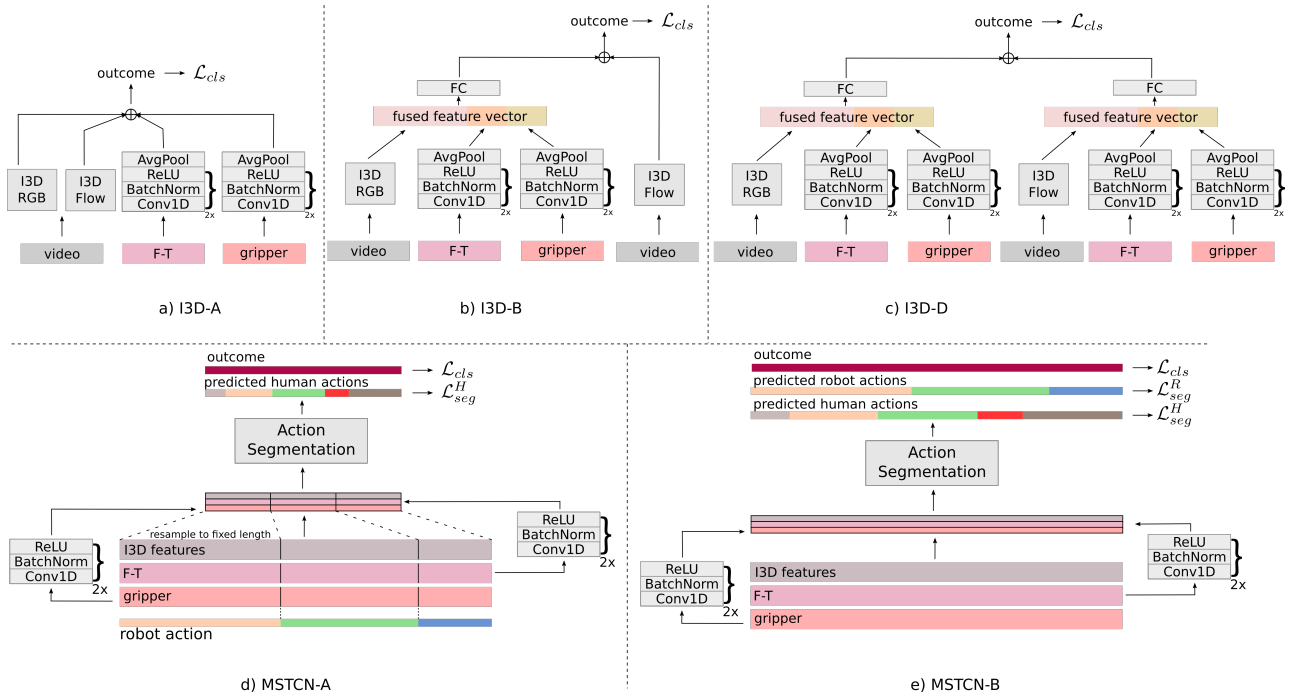


Fig. 3. a) **I3D-A**: I3D network with late fusion of modalities. b) **I3D-B**: I3D network with intermediate fusion of F-T and gripper features with RGB stream. **I3D-C**, which is not shown, is very similar to I3D-B, but the intermediate fusion of F-T and gripper features is with the Flow instead of the RGB stream. c) **I3D-D**: I3D network with intermediate fusion of F-T and gripper features with RGB and Flow streams. d) **MSTCN-A**: MSTCN network that takes I3D, F-T and gripper features as input and predicts the human actions for each frame and the outcome. The input is resampled to a fixed length based on the robot actions. e) **MSTCN-B**: MSTCN network with original-length inputs and an additional prediction head for the robot actions.

person’s arm in the video<sup>5</sup>. The outcome of the trial is one of the classes listed in Table II. The robot (Toyota Human Support Robot (HSR) or Kinova Gen3) and the task (R2H or H2R) are included as metadata for each trial.

#### IV. BASELINE APPROACHES

The input data consists of a sequence of  $T$  video frames  $x_{1:T} = (x_1, \dots, x_T)$ , force-torque measurements  $ft_{1:T} = (ft_1, \dots, ft_T)$ <sup>6</sup> and the state of the gripper  $g_{1:T} = (g_1, \dots, g_T)$ , where  $g_t \in \{-0.5, 0.0, 0.5\}$  refers to {open, partially closed, and closed}, respectively. Additional annotations are the current human action  $h_{1:T} = (h_1, \dots, h_T)$  being performed, where  $h_t \in \{\text{idle, approach, interact, retract, post-idle, not released, dropped}\}$  and which is only available during training, and the current robot action  $r_{1:T} = (r_1, \dots, r_T)$  being performed, where  $r_t \in \{\text{approach, interact, retract}\}$  and which is available during training and inference time. The task is to classify the outcome of the handover action  $o \in \{\text{success, no approach, no grasp, drop, no release}\}$ , where *no grasp* only applies to R2H and *no release* only applies to H2R. We present two baseline methods for detecting the outcome of the handover action.

##### A. Video classification

For the first baseline, we propose multi-modal variants of the pre-trained Inflated 3D ConvNet (I3D) [20] network. We investigate four variants, which are illustrated in Fig. 3:

- I3D-A**: networks to classify the outcome from video (with separate RGB and optical flow streams), F-T and gripper state are trained separately, and their outputs are combined with late fusion.
- I3D-B**: intermediate fusion and a fully-connected (FC) layer are used to combine the features of the RGB stream, F-T and gripper, whose output is combined with the output of the optical flow stream with late fusion.
- I3D-C**: intermediate fusion and a FC layer are used to combine the features of the optical flow stream, F-T and gripper, whose output is combined with the output of the RGB stream with late fusion.
- I3D-D**: intermediate fusion is used to combine features of the F-T and gripper streams with both RGB and optical flow features, before applying late fusion to combine the two outputs.

For the video stream, the input consists of 64 frames that are sampled at equal intervals from the entire video of a trial. F-T measurements are normalized based on the standard deviation and mean of measurements in the training set.

##### B. Human action segmentation

The failures in the dataset are caused either by the person not performing the expected action (such as not reaching out) or performing an unexpected action (such as dropping the object). Therefore, we consider the auxiliary task of classifying the actions of the human as a means of detecting unperformed and unexpected actions. Recent approaches for temporal action segmentation in untrimmed videos make use of temporal convolutions [21], [22], [23]

<sup>5</sup>Although all annotations were performed by a single annotator, there might still be some ambiguity in the exact boundaries of the person’s actions.

<sup>6</sup> $ft$  is a vector consisting of the force  $(f_x, f_y, f_z)$  and torque  $(\tau_x, \tau_y, \tau_z)$  in three directions expressed in Newton and Newton-meter, respectively.

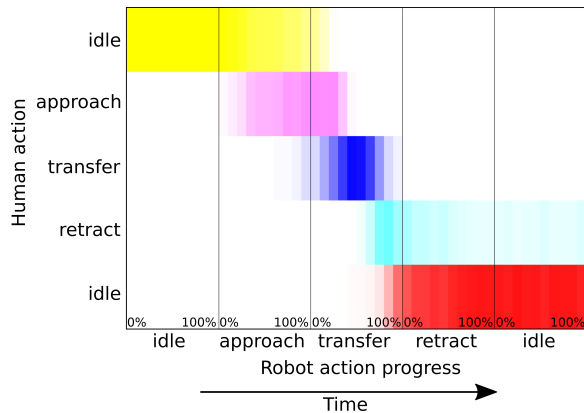


Fig. 4. Frequency of each human action corresponding to the robot’s actions in the training set for nominal R2H handovers. The progress of each robot action from 0% - 100% is discretized into 10 bins

or transformer-based architectures [24]. We select the multi-stage temporal convolutional network (MS-TCN) [22] due to its good performance on action segmentation datasets and since transformer-based methods require larger datasets. Although improved versions of MS-TCN exist [25], we use the basic architecture here for the baseline.

MS-TCN uses dilated temporal convolutions on a sequence of clip-level features to perform action segmentation. Each stage of the network outputs a prediction, which is refined by the next stage. We use a 2-stage MS-TCN network for segmenting the actions of the human and extend it to the multi-modal setting. RGB and optical flow features are extracted for each frame from pre-trained I3D networks using a 64-frame window from the past, resulting in a feature vector of length 2048 for each frame. F-T and gripper state are passed through 1D convolutional layers before being concatenated to the I3D features. Therefore, the input to the segmentation network is a sequence of feature vectors  $f_{1:T} = (f_1, \dots, f_T)$ , representing the video, F-T and gripper state. The target for the network is the sequence of human actions  $h_{1:T}$  and the overall outcome  $o$ . The predicted overall outcome is output as a sequence  $\hat{o}_{1:T}$ , although only  $\hat{o}_T$  is considered at test time.

Fig. 4 shows histograms for each human action corresponding to the robot action being performed in nominal R2H handovers (from the training set only). It can be seen, for example, that the person typically approaches the robot when the robot is approaching the person and during the initial stages of *transfer*. In order to exploit this correlation, we consider two ways in which the robot action can be used to improve the classification of the human action (see Fig. 3):

- (a) **MSTCN-A**: the input features are resampled such that the features corresponding to each robot action are represented by a fixed length sequence.
- (b) **MSTCN-B**: the original sequence length is used, but the network predicts the robot action in addition to predicting the human action and overall outcome.

1) *Loss functions*: The cross entropy loss is used for the classification loss of the outcome,  $\mathcal{L}_{cls} = -\log(\hat{o}_c)$ , where  $\hat{o}_c$  is the predicted softmax score for the ground-truth class  $c$ . For the segmentation network, the classification loss for the

outcome is summed over all frames  $t$  and all stages  $s$  of the multi-stage network, so that  $\mathcal{L}_{cls} = -\sum_s \frac{1}{T} \sum_t \log(\hat{o}_c^{s,t})$ .

The segmentation loss (Eq. 3) consists of the classification loss of the robot or human action (Eq. 1), and a smoothing loss (Eq. 2) which reduces over-segmentation [22]. The smoothing loss is weighted by  $\lambda = 0.15$  and we use  $\tau = 4$  as in [22]. The human action and robot action segmentation loss terms,  $\mathcal{L}_{seg}^H$  and  $\mathcal{L}_{seg}^R$ , are also summed over all stages  $s$  of the multi-stage network.  $\hat{y}_c^{s,t}$  is the probability of class  $c$  at time  $t$  and stage  $s$ , where  $c$  corresponds to the ground-truth human or robot action, respectively.

$$\mathcal{L}_{segcls} = -\sum_s \frac{1}{T} \sum_t \log(\hat{y}_c^{s,t}) \quad (1)$$

$$\mathcal{L}_{smooth} = \sum_s \frac{1}{TC} \sum_{t,c} \min(|\log(\hat{y}_c^{s,t-1}) - \log(\hat{y}_c^{s,t})|, \tau)^2 \quad (2)$$

$$\mathcal{L}_{seg} = \mathcal{L}_{segcls} + \lambda \mathcal{L}_{smooth} \quad (3)$$

2) *Metrics*: All networks are trained five times, and we report the mean and standard deviation of the classification accuracy for the outcome and the segmentation metrics, which are frame-wise accuracy and segmental F1-score [21] at overlapping thresholds of 10%, 25% and 50%.

## V. RESULTS

The main results are shown in Table III, in which the networks are trained using different combinations of input modalities and loss functions. For I3D-A with individual modalities (1-3), the video modality performs better than F-T or gripper alone. In terms of fusion strategies (4-7), I3D-D, in which F-T and gripper features are combined using intermediate fusion with both RGB and flow features, performs best. It also outperforms the uni-modal variants. It needs to be noted that for a single modality, all I3D variants are the same. When we analyze the impact of each modality (8-12), we observe that the combination of video and F-T performs best. We next investigate multitask learning with MSTCN by considering only video as input (13-17). We observe a clear improvement when  $\mathcal{L}_{seg}^H$  is added as a second task. Overall, MSTCN-B with all three tasks performs best. It also performs better than the video-only I3D-A model (1). In the case of multimodal inputs (18-24), both MSTCN variants show the best performance with all three modalities, with a marginal drop in performance when using the robot action segmentation loss term for MSTCN-B (24). The latter is not surprising since the additional modalities provide information about the robot action such that predicting the robot action is not a challenging task anymore. We also note that the video + F-T combination (18, 21) performs better compared to video + gripper (19, 22), as with I3D-D (9, 10).

### A. Action Segmentation

Table IV shows the human action segmentation metrics. We also include a baseline which is not learned, but is derived from the correlation between human and robot actions (Fig. 4), by selecting the most likely human action

TABLE III  
IMPACT OF MODALITIES, LOSS TERMS AND NETWORK

ID	Model	Modality			Loss function			Accuracy
		V	F-T	G	$\mathcal{L}_{cls}$	$\mathcal{L}_{seg}^H$	$\mathcal{L}_{seg}^R$	
1	I3D-A	✓			✓			<b>64.8 ± 3.3</b>
2	I3D-A		✓		✓			29.4 ± 6.1
3	I3D-A			✓	✓			38.2 ± 7.5
4	I3D-A	✓	✓	✓	✓			65.4 ± 2.2
5	I3D-B	✓	✓	✓	✓			66.5 ± 3.0
6	I3D-C	✓	✓	✓	✓			64.1 ± 1.4
7	I3D-D	✓	✓	✓	✓			<b>67.9 ± 3.3</b>
8	I3D-D	✓			✓			64.8 ± 3.3
9	I3D-D	✓	✓		✓			<b>69.1 ± 1.5</b>
10	I3D-D	✓		✓	✓			67.3 ± 2.6
12	I3D-D	✓	✓	✓	✓			67.9 ± 3.3
13	MSTCN-A	✓			✓			57.9 ± 1.8
14	MSTCN-A	✓			✓	✓		63.8 ± 4.8
15	MSTCN-B	✓			✓			56.6 ± 2.2
16	MSTCN-B	✓			✓	✓		62.7 ± 3.2
17	MSTCN-B	✓			✓	✓	✓	<b>67.8 ± 3.7</b>
18	MSTCN-A	✓	✓		✓	✓		67.3 ± 2.3
19	MSTCN-A	✓		✓	✓	✓		65.6 ± 2.0
20	MSTCN-A	✓	✓	✓	✓	✓		<b>71.4 ± 1.8</b>
21	MSTCN-B	✓	✓		✓	✓		69.0 ± 2.7
22	MSTCN-B	✓		✓	✓	✓		66.4 ± 2.8
23	MSTCN-B	✓	✓	✓	✓	✓		<b>71.1 ± 2.6</b>
24	MSTCN-B	✓	✓		✓	✓	✓	70.7 ± 1.8

V: Video/I3D features, F-T: force-torque, G: gripper

TABLE IV  
SEGMENTATION METRICS

Model	F1@{10,25,50}			Frame-wise acc
<b>Baseline using correlation between robot and human action</b>				
Correlation	58.1	48.6	30.8	44.5
<b>I3D features only</b>				
MSTCN-A	74.7 ± 1.4	68.0 ± 1.3	51.1 ± 1.9	70.5 ± 2.6
MSTCN-B	67.9 ± 2.9	62.7 ± 3.0	48.2 ± 2.2	69.6 ± 2.3
<b>I3D features, F-T and gripper</b>				
MSTCN-A	80.3 ± 0.5	74.1 ± 0.7	57.1 ± 2.3	75.9 ± 1.1
MSTCN-B	70.8 ± 2.0	65.2 ± 2.1	49.7 ± 2.6	73.2 ± 0.8
<b>I3D features, F-T and gripper (only <math>\mathcal{L}_{seg}^H</math>)</b>				
MSTCN-A	74.7 ± 2.2	70.7 ± 1.9	56.7 ± 1.5	76.6 ± 1.5
MSTCN-B	73.5 ± 2.4	69.3 ± 3.0	56.6 ± 2.4	77.0 ± 0.9

given the current robot action and progress from the relative occurrences in the training set. We observe that MSTCN-A is able to better segment human actions, with increased performance for both variants with multimodal inputs.

### B. Robot Generalization

We train the networks using only data from one robot, and evaluate on both to investigate how well the networks are able to generalize to a new robot. Since F-T and gripper data are represented similarly for both platforms, only the viewpoint, appearance and motions of the robot are the factors which differ between the two. The results are shown in Table V. The performances in all cases are lower than when trained with the full dataset, but we note that the

TABLE V  
ROBOT GENERALIZATION (TRAIN→TEST)

Model	T→T	T→K	K→T	K→K
I3D-D	51.1 ± 4.1	43.0 ± 4.4	55.1 ± 5.5	62.2 ± 4.6
MSTCN-A	70.1 ± 2.2	56.3 ± 3.2	47.6 ± 6.6	48.9 ± 3.0
MSTCN-B	70.5 ± 2.8	49.3 ± 4.6	44.7 ± 2.0	56.3 ± 4.3

T: Toyota HSR, K: Kinova Gen3

TABLE VI  
SEPARATELY TRAINED R2H AND H2R NETWORKS

Model	R2H	H2R
I3D-D	67.2 ± 3.3	61.5 ± 5.8
MSTCN-A	69.0 ± 5.9	65.2 ± 6.0
MSTCN-B	68.5 ± 3.4	75.3 ± 4.4

MSTCN variants are able to generalize better when trained on the larger HSR dataset, whereas I3D-D generalizes better when trained on the smaller Kinova dataset.

### C. Effect of training on tasks separately

All previous experiments jointly trained on R2H and H2R. When the networks are trained on each task separately (see Table VI) the results are comparable, but slightly lower on average compared to training on both tasks together.

### D. Summary

Overall, the results show that while video is an essential modality for detecting these failures, the addition of F-T in particular, improves the performance. Adding the robot action segmentation loss for MSTCN-B improves the performance in the unimodal setting, but not in the multimodal setting. I3D-D, MSTCN-A and MSTCN-B all perform well for the failure detection task, whereas MSTCN-A shows a better performance overall with better segmentation accuracy. In terms of interpretability of the results, the predicted human actions with the MSTCN networks can be used to analyze the reasons for the failures and when they occur; however this needs to be investigated further in future work.

## VI. LIMITATIONS

The handovers in the dataset, including the failures, are performed, and as such may not represent natural handovers or failures; however it is hard to capture naturally occurring failures. The baseline approaches for failure detection only classify the outcome after the complete trial. However, a robot should react to a failure as soon as it has occurred, for example, by prompting the person if they have not approached. Slight modifications of the MSTCN architecture would allow for causal predictions by applying the temporal convolutions only on past inputs, thus enabling online prediction of the outcome.

## VII. CONCLUSIONS

In order to address the need for datasets and evaluation methods which consider human-induced failures during handovers, we presented a multimodal handover failure detection dataset containing both R2H and H2R handovers. The data consists of video, force-torque and robot joint states of both successful and failed handovers. Each video is annotated with the robot's and person's actions during the handover and the overall outcome. Our baseline approaches show that video is an essential modality for detecting failures and using multimodal data, in particular the force-torque sensor, is beneficial. Using the human action segmentation approach to detect failures online, and to interpret them based on the human's actions, such that the robot can respond appropriately, are an interesting direction for future research.

## REFERENCES

- [1] S. Parastegari, E. Noohi, B. Abbasi, and M. Žefran, “Failure Recovery in Robot–Human Object Handover,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 660–673, 2018.
- [2] M.-J. Davari, M. Hegedus, K. Gupta, and M. Mehrandezh, “Identifying Multiple Interaction Events from Tactile Data during Robot–Human Object Transfer,” in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–6.
- [3] R. Liu, R. Chen, and C. Liu, “Task-Agnostic Adaptation for Safe Human-Robot Handover,” *arXiv preprint arXiv:2209.09418*, 2022.
- [4] A. G. Eguiluz, I. Rañó, S. A. Coleman, and T. M. McGinnity, “Reliable object handover through tactile force sensing and effort control in the Shadow Robot hand,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 372–377.
- [5] W. Yang, C. Paxton, A. Mousavian, Y.-W. Chao, M. Cakmak, and D. Fox, “Reactive Human-to-Robot Handovers of Arbitrary Objects,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3118–3124.
- [6] M. Mavsar and A. Ude, “RoverNet: Vision-Based Adaptive Human-to-Robot Object Handovers,” in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 858–864.
- [7] Y. L. Pang, A. Xompero, C. Oh, and A. Cavallaro, “Towards safe human-to-robot handovers of unknown containers,” in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 51–58.
- [8] S. Thoduka and N. Hochgeschwender, “Benchmarking Robots by Inducing Failures in Competition Scenarios,” in *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. AI, Product and Service: 12th International Conference, DHM 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part II*. Springer, 2021, pp. 263–276.
- [9] Z. Xu, S. Escalera, A. Pavão, M. Richard, W.-W. Tu, Q. Yao, H. Zhao, and I. Guyon, “Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform,” *Patterns*, vol. 3, no. 7, p. 100543, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666389922001465>
- [10] E. C. Grigore, K. Eder, A. G. Pipe, C. Melhuish, and U. Leonards, “Joint Action Understanding improves Robot-to-Human Object Handover,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4622–4629.
- [11] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić, “Object Handovers: a Review for Robotics,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1855–1873, 2021.
- [12] S. Parastegari, E. Noohi, B. Abbasi, and M. Žefran, “A fail-safe object handover controller,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2003–2008.
- [13] P. Rosenberger, A. Cosgun, R. Newbury, J. Kwan, V. Ortenzi, P. Corke, and M. Grafinger, “Object-Independent Human-to-Robot Handovers Using Real Time Robotic Vision,” *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 17–23, 2020.
- [14] Z. Han and H. A. Yanco, “Reasons People Want Explanations After Unrecoverable Pre-Handover Failures,” in *ICRA Workshop on Human-Robot Handovers*, 2020. [Online]. Available: <https://arxiv.org/abs/2010.02278>
- [15] C. Meng, T. Zhang, and T. lun Lam, “Fast and Comfortable Interactive Robot-to-Human Object Handover,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3701–3706.
- [16] I. Mamaev, D. Kretsch, H. Alagi, and B. Hein, “Grasp Detection for Robot to Human Handovers Using Capacitive Sensors,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 552–12 558.
- [17] F. Iori, G. Perovic, F. Cini, A. Mazzeo, E. Falotico, and M. Controzzi, “DMP-Based Reactive Robot-to-Human Handover in Perturbed Scenarios,” *International Journal of Social Robotics*, pp. 1–16, 2023.
- [18] M. Mohandes, B. Moradi, K. Gupta, and M. Mehrandezh, “Robot to Human Object Handover Using Vision and Joint Torque Sensor Modalities,” in *International Conference on Robot Intelligence Technology and Applications*. Springer, 2022, pp. 109–124.
- [19] Y.-W. Chao, C. Paxton, Y. Xiang, W. Yang, B. Sundaralingam, T. Chen, A. Murali, M. Cakmak, and D. Fox, “HandoverSim: A Simulation Framework and Benchmark for Human-to-Robot Object Handovers,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6941–6947.
- [20] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.
- [21] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.
- [22] Y. A. Farha and J. Gall, “MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3575–3584.
- [23] S. Li, Y. A. Farha, Y. Liu, M.-M. Cheng, and J. Gall, “MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6647–6658, 2023.
- [24] F. Yi, H. Wen, and T. Jiang, “ASFormer: Transformer for Action Segmentation,” *British Machine Vision Conference (BMVC)*, 2021.
- [25] G. Ding, F. Sener, and A. Yao, “Temporal Action Segmentation: An Analysis of Modern Techniques,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.