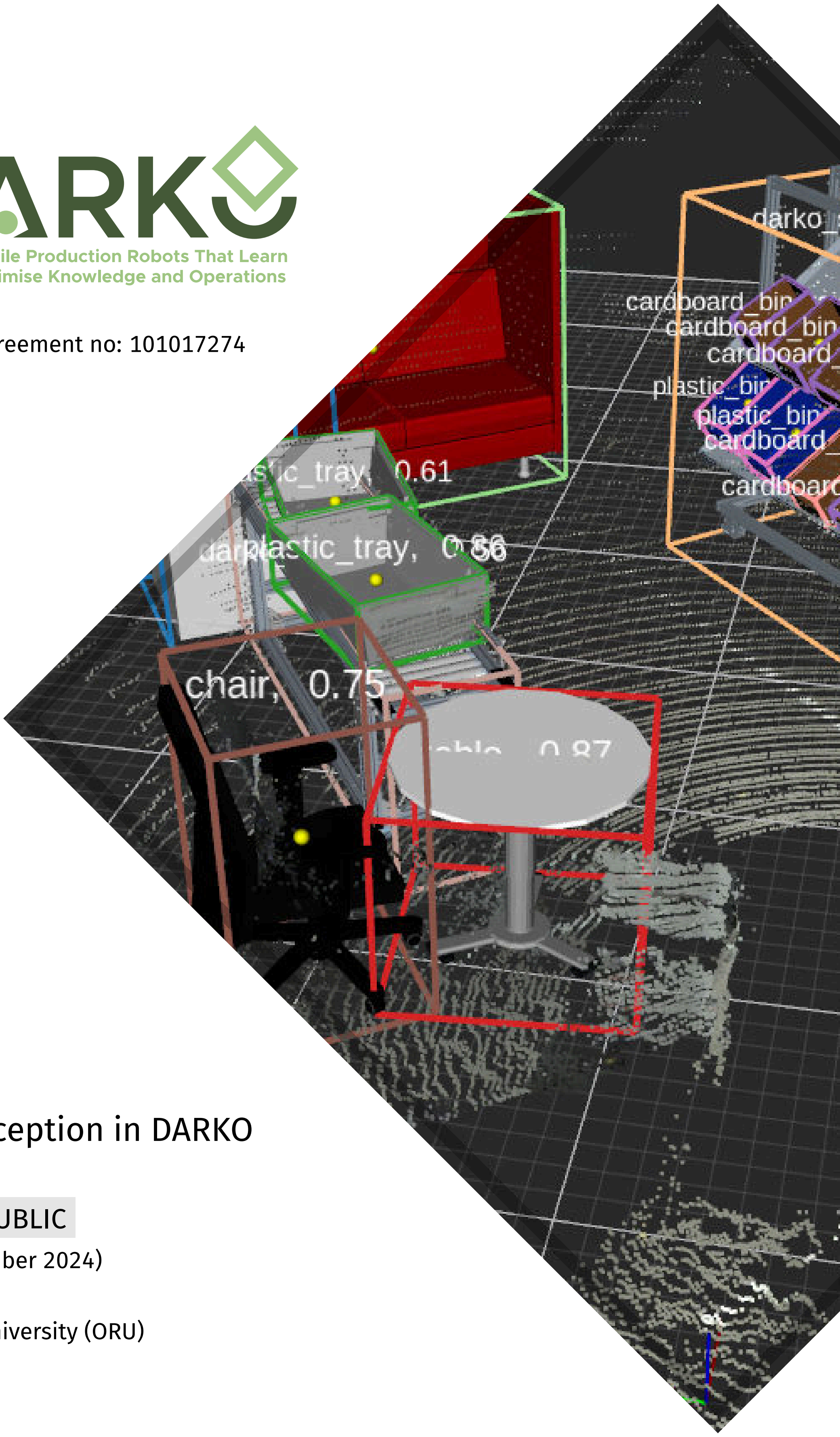
DARKO

Dynamic Agile Production Robots That Learn and Optimise Knowledge and Operations

H2020-ICT-2020-2 Grant agreement no: 101017274



DELIVERABLE 2.3
Final report on perception in DARKO

Dissemination Level: PUBLIC

Due date: month 48 (December 2024)

Deliverable type: Report

Lead beneficiary: Örebro University (ORU)

Contents

List of Abbreviations

Abbreviation	Meaning
9-DoF	9 degrees of freedom (3D position, 3D orientation, 3D extents)
API	Application Programming Interface, the public interface provided by a library for use by software developers
ARENA 2036	A large research campus in form of a modern factory hall in Stuttgart-Vaihingen, Germany. Provides an innovation platform for mobility & production of the future and hosts DARKO project demonstrations.
CAD	Computer Aided Design
CNN	Convolutional Neural Network
DLA	Dynamic Layer Aggregation backbone
CUDA	Compute Unified Device Architecture for hardware acceleration of neural network computations on Nvidia GPUs
DeTR	Detection Transformer, transformer-based NN for object detection
FOV	Field of view of a sensor
GigE	Gigabit Ethernet
GPU	Graphics Processing Unit, used as a deep learning accelerator to train and run inference on neural networks
GRU	Gated Recurrent Unit, a gating mechanism in RNNs
KLT	Kleinladungsträger, a standardized plastic box in different sizes often used in intralogistics.
LiDAR	Light Detection And Ranging, a time-of-flight-based sensor that produces point clouds. Also spelled “lidar”.
MPC	Model-Predictive Control
NN	Neural Network
NIC	Network Interface Controller
ONNX	Open Neural Network Exchange
ORU	Örebro University, member of the DARKO consortium
PTP	Precision Time Protocol, a more accurate clock synchronization protocol than Network Time Protocol (NTP)
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue, Depth
RNN	Recurrent Neural network
ROS	Robot Operating System, see www.ros.org
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
SPENCER	EU FP7 project (2013–2016) which deployed a mildly humanized service robot in a busy airport terminal at Amsterdam Schiphol Airport.
SVM	Support Vector Machine, a machine learning classifier
ToE	Trigger-over-Ethernet, method for camera triggering
UNIPI	Università di Pisa, member of the DARKO consortium
VRAM	Video memory of a GPU that can be used to deploy neural networks
WP	Work package
YOLO	A series of 2D object detectors developed by J. Redmon

1 Introduction

This deliverable reports on the final perception system developed in the EU H2020 project DARKO in work package WP2 *3D Perception and Scene Understanding*.

The preceding Deliverable D2.2 reported on the initial work done towards the perception system until Month 30 (June 2023).Within this deliverable, we refer to the contents of the earlier D2.2 where relevant, to avoid duplicate content. The present deliverable focuses on new developments that have happened in the final period of the project until Month 48 (December 2024), additionally incorporating also some late-breaking insights from the MS4 final stakeholder meeting and demonstration in month 54 (June 2025).

1.1 Relation to DARKO objectives

The work in WP2 and the perception components described in this deliverable contribute to all of DARKO's scientific objectives O1–O4 and through WP8 we have integrated the perception system to demonstrate feasibility (O5).

T2.1–T2.4 address **O1: Efficiency in manipulation**. T2.2–T2.3 focus on perception for manipulation, including in-hand grasp perception and perception of thrown objects. T2.1 (object-level semantics) includes work on perceiving the boxes required for picking and placing objects thus defining the area of interest for T2.2 (perception for manipulation). In particular, T2.1 has contributed efficient 360-degree object detection from fish-eye camera and lidar data; T2.2 contributes methods for learning to grasp objects in difficult configurations, including ways to make reinforcement learning in sparse reward settings more efficient; T2.3–T2.4 are important for enabling throwing, which makes the mobile manipulation system more efficient.

T2.5 addresses **O2: Efficiency in human–robot co-production**; specifically, real-time 3D detection and tracking of humans and their articulated 3D poses from onboard fish-eye RGB cameras alone.

T2.1 and T2.2 address **O3: Efficient deployment**; in particular by investigating methods to reduce the manual labeling effort to enable efficient deployment in environments underrepresented in large-scale public data sets (where pretrained object detection methods fall short), considering which deep neural network architecture to use under data and hardware constraints, and developing methods for grasping novel objects without prespecified grasp configurations.

T2.3 addresses **O4: Risk-aware operation for safety and efficiency** by assessing the grasp configuration before attempting to throw, and in particular methods for detecting and avoiding objects slipping from the hand. T2.5 further contributes to this objective by providing the necessary perception of humans for the risk- and human-awareness developed the remaining work packages.

1.2 Relation to other work packages

With perception being located at the beginning at the *sense – plan – act* chain, the functionalities presented in this deliverable provide functionalities that are crucial for the DARKO robot platform to successfully achieve its overall objectives, by enabling the robot to robustly perceive and understand its 3D surroundings in an intralogistics and agile production environment of the factory of the future.

Figure 1 illustrates the relation of work package WP2, which this deliverable reports on, to the other technical work packages in DARKO. Either through direct communication, or indirectly through the mapping and localization work package (WP3), WP2 provides essential cues to dynamic manipulation (WP4), human-robot spatial interaction (WP5), motion planning for the mobile base (WP6) and risk management and scheduling (WP7).

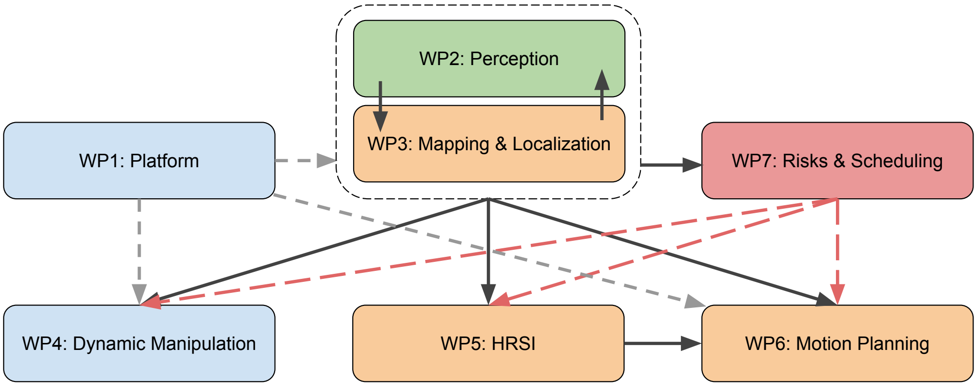


Figure 1: Relation of WP2, which this deliverable reports on, to other work packages in DARKO. WP2 is shown in green on top. It provides input both to work packages related to mobile manipulation (blue), as well as navigation in shared environments (yellow); here it is closely intertwined with WP3 (mapping and localization), which temporally integrates static and dynamic observations from tasks T2.1 and T2.5. Black arrows denote data flow during operation, dashed red arrows indicate constraints and orchestration, and dashed grey arrows indicate hardware dependencies.

1.3 Key highlights

The key highlights of the DARKO perception system are:

1. Novel data-efficient methods for object-level semantics and 3D scene graph prediction. These include an extension of the real-time capable TR3D method [20] for 9DoF object detection, and 360° perception pipeline development, leveraging the adapted TR3D to develop a full surround object detection pipeline, effectively utilizing the robot's fisheye camera setup. We have also made advancements in open-vocabulary 3D scene understanding and relational modeling: including methods for predicting open-vocabulary 3D scene graphs and for modeling object relationships within neural radiance fields [1, 2], and using 3D scene graphs and VLMs for open-vocabulary embodied tasks, exemplified by the GraphEQA [3] framework.

2. Data-efficient learning-based methods for skeleton-based human activity recognition; for full-body 3D pose prediction with the aim of bridging temporary occlusions; and an extensive cross-modal comparison of human detection methods in intralogistics. These include a novel Transformer-based approach for prediction of articulated 3D human poses (and trajectories), a data-efficient multi-class human pose classifier that translates articulated 3D body poses into qualitative class labels (e. g., standing, sitting), extending the 3D human pose estimator MeTRAbs to natively support fisheye images (Fisheye-MeTRAbs), and integrating a state-of-the-art skeleton-based multi-frame activity recognition approach for detecting dynamic gestures that are relevant for intralogistics scenarios.

3. Novel, comprehensive domain-specific datasets and efficient annotation strategies e. g. for 3D object detection and wide-FOV 3D human pose estimation targeted specifically at DARKO use-cases in intralogistics scenarios, which are so far underrepresented in publicly available datasets. This includes a novel multi-modal,

multi-view dataset for 3D human pose estimation using fisheye cameras and 3D lidar in industrial scenarios, along with a complex multi-sensor recording setup for accurate groundtruth acquisition of 3D body poses. Furthermore, an efficient multi-view labeling strategy based on SLAM to facilitate the creation of custom datasets for training of 3D object detectors.

4. Novel methods for perception for manipulation based upon direct grasp pose estimation and a learning-based combination with manipulation primitives, with real-world experiments using authentic objects from the DARKO target use-case. These include the VoteGrasp method for grasp pose estimation, directly regressing grasp poses from RGB-D sensor data [4]; the ED-PMP method for hierarchical reinforcement learning, learning control policies for parameterised sub-tasks to improve training efficiency (specifically for picking up objects with "occluded grasps" with the help of external surfaces); and a novel apprach to improve learning efficiency for reinforcement learning with sparse rewards (KEA [5]), which makes the design of methods that use RL to grasp novel objects in difficult configurations much easier.

5. Novel approaches for in-hand perception to estimate in-hand object pose, to predict grasp failures with soft robotic hands and avoid object slippage, and to generate optimal manipulation patterns to collect information about grasped object's inertia.

6. Integration with downstream applications like manipulation, semantic SLAM and semantic-aware navigation. This includes porting of ROS nodes for human and object perception to ONNX and TensorRT backends for significantly faster and more memory-efficient inference on embedded Nvidia Jetson Orin hardware.

2 Perception System Overview

The final perception system on the DARKO robot is the result of an iterative design process aimed at achieving robust, 360° semantic understanding of objects and humans. This section provides an overview of the final hardware and compute setup, followed by a summary of the extensive calibration efforts required to fuse the multi-modal sensor data. The concrete perception modules for objects, humans and manipulation are discussed in the subsequent sections of this document.

2.1 Hardware and Compute Setup

The robot is equipped with a comprehensive sensor suite and a distributed computing architecture to handle the demanding perception tasks. The primary sensors for broader scene understanding (T2.1 and T2.5) are an **Ouster OS0-128 3D lidar** and two back-to-back 220-degree **fisheye cameras**. The initial sensor configuration featuring two 180-degree fisheye cameras placed next to each other was updated to a vertically stacked arrangement of two 220-degree cameras, as shown in fig. 3. This vertical stacking optimizes the observable field of view, as it minimizes occlusions from Ethernet cables that would be more prominent in a horizontal layout, while the robot's base and the lidar on top of the mast already occlude the vertical boundaries.

Additionally, the robot is equipped with two **Azure Kinect RGB-D cameras** for perception for manipulation (T2.2–T2.4), with the top one being mounted on a **pan-tilt unit**, and the bottom one always facing sideways to be able to observe the shelf for picking. However, due to their limited field of view (FOV) and the lack of mechanical stability of the pan-tilt unit installed by the robot manufacturer (making an accurate extrinsic calibration nearly impossible), these RGB-D cameras were not used for the final surround view perception of broader-level scene understanding in tasks T2.1 and T2.5.

The **compute hardware** consists of two main units, as depicted in Figure 2: a Neousys Nuvo industrial PC with an NVIDIA RTX 3060 GPU and a Jetson AGX Orin. In the final phase of the project, we found that the industrial PC's CPU was a bottleneck. To address this, we installed an additional external 120 mm fan, which allowed us to increase the CPU's TDP from 35W to 45W, providing approximately 25% more processing power. The Jetson AGX Orin is primarily used to run the Azure Kinect drivers and RViz for visualization, as its shared CPU/GPU memory architecture proved to be more efficient for these tasks, reducing latency from CPU/GPU data transfers. To enable efficient data streaming between the two computers, we implemented a ROS-based RTSP image transport based upon [21] and Nvidia hardware codec libraries, which was essential for transmitting the fisheye camera images from the Nuvo PC to the Jetson without congesting the network and interfering with other critical components like the Franka robot control interface.

Finally, an Arduino-based microcontroller was added to the system to provide a hardware-level synchronization signal, simultaneously triggering both GigE fisheye cameras at a frame rate of 15 Hz.

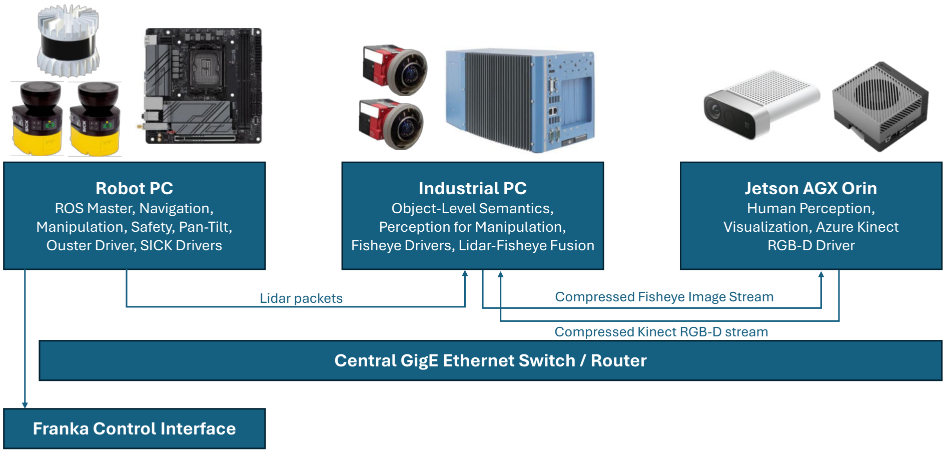

Robot PC
ROS Master, Navigation, Manipulation, Safety, Pan-Tilt, Ouster Driver, SICK Drivers
Industrial PC
Object-Level Semantics, Perception for Manipulation, Fisheye Drivers, Lidar-Fisheye Fusion
Jetson AGX Orin
Human Perception, Visualization, Azure Kinect RGB-D Driver
Lidar packets
Compressed Fisheye Image Stream
Compressed Kinect RGB-D stream
Central GigE Ethernet Switch / Router
Franka Control Interface

Figure 2: Sensor and compute setup on the DARKO robot. Except for the two SICK safety laser scanners and the Ouster lidar, which communicate via the central ethernet switch, all sensors are connected directly to the respective computers (via USB3 for Azure Kinect, and a separate GigE card for the fisheye cameras, which are triggered synchronously by a separate Arduino-based microcontroller).

2.2 Sensor Calibration

Accurate sensor calibration is fundamental for fusing data from multiple sensors. Our efforts focused on precisely calibrating the intrinsic parameters of the fisheye cameras and the extrinsic transformations between all cameras and the lidar.

As part of the research in T2.5, we identified the **Double Sphere camera model** [22] as highly suitable for the wide-angle fisheye lenses used on the DARKO robot, as it provides a good balance between accuracy and computational efficiency, including a closed-form analytical inverse projection. For the **intrinsic calibration** of the fisheye lenses and the **extrinsic calibration between the two cameras**, we used a commercial tool (calib.io [23]) with an AprilGrid calibration target, as shown in Figure 4.

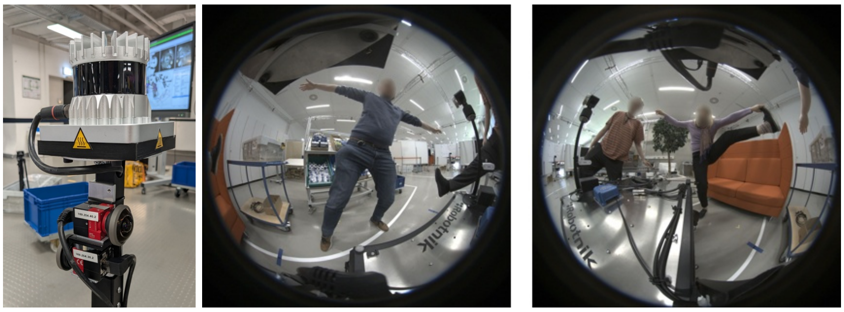


Figure 3: *Left:* Final sensor setup for broader-level scene understanding in T2.1 and T2.5, consisting of two stacked 220-degree fisheye cameras and an Ouster OS0-128 3D lidar. *Right:* Exemplary images from the forward- and backward-facing fisheye cameras, which together with the lidar facilitate surround-view 3D perception.

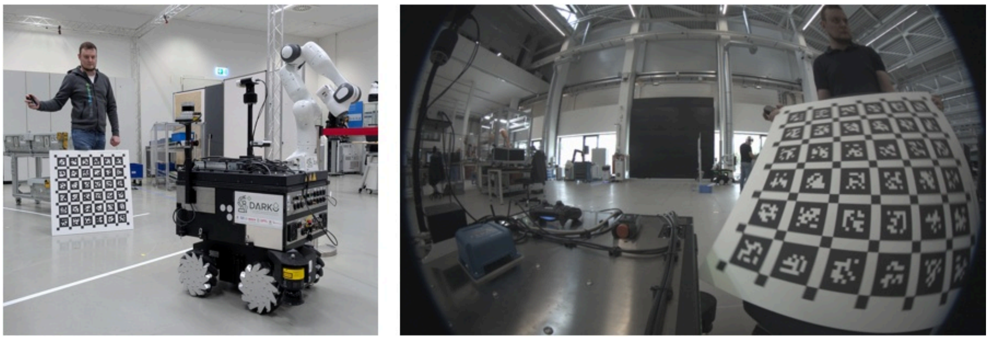


Figure 4: For intrinsic and extrinsic camera-to-camera calibration, we use an AprilGrid calibration target in combination with a commercial calibration software toolkit.

For **lidar-to-camera** extrinsic calibration, we utilized a direct visual lidar calibration tool available as a ROS package [24], which we extended to support the Double Sphere camera model. We achieved the most reliable results through manual feature matching, where an operator clicks on corresponding points in the fisheye images and the lidar point cloud. To facilitate this process, we enhanced the tool's GUI with options for hiding the ceiling and other visualization improvements.

To improve the **quality of the calibration images**, particularly by minimizing motion blur, we developed a speech-guided image acquisition approach [6]. As shown in the flow diagram in Figure 5, this system allows an operator to trigger image capture using voice commands, freeing their hands to hold the calibration target steady (Figure 6). This method proved highly effective in acquiring sharp, high-quality images, which are essential for achieving a successful and accurate calibration convergence (Figure 7).

Despite these advancements, an open challenge remains with the 220-degree fisheye lenses: the detection of AprilTags at very close ranges (<40 cm) is unreliable. While some existing solutions have been proposed in the literature, they did not work in our specific setup. Bosch is continuing to investigate this problem in an ongoing collaboration with RWTH Aachen University.

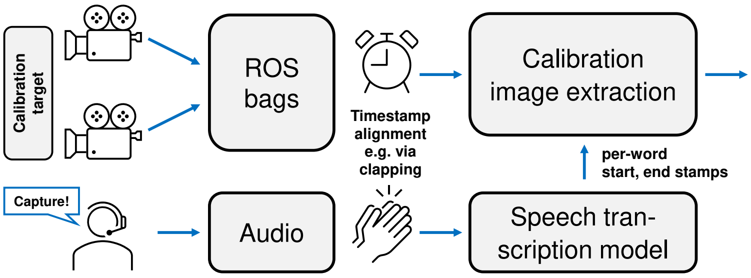
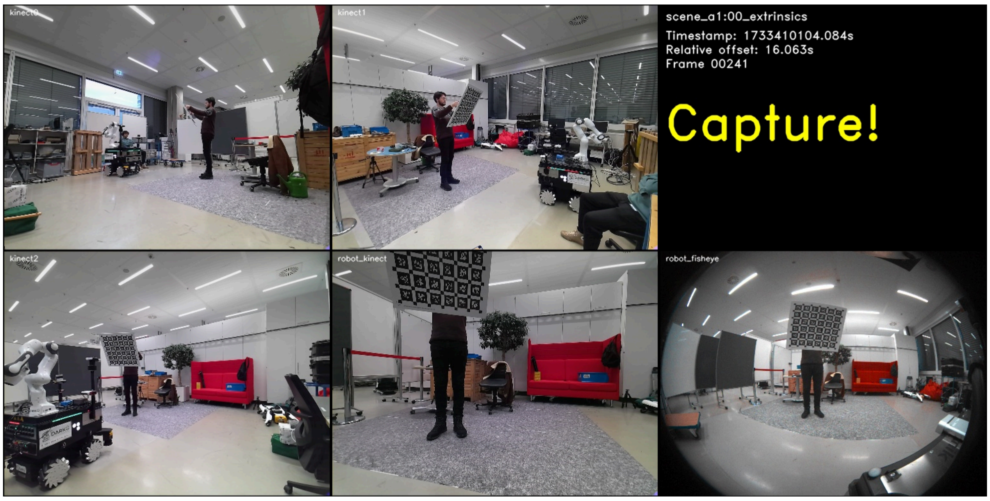

Calibration target
ROS bags
Timestamp alignment e.g. via clapping
Calibration image extraction
per-word start, end stamps
Capture!
Audio
Speech transcription model

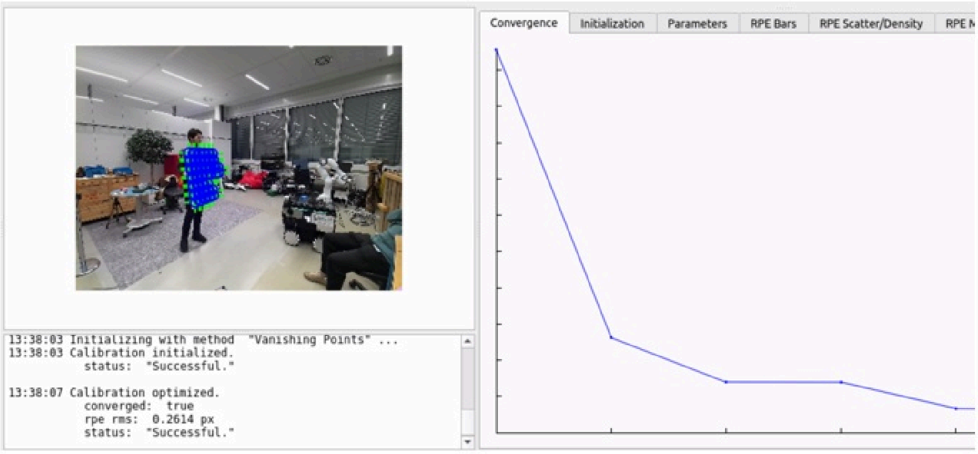
Figure 5: Flow diagram of the proposed approach for speech-guided calibration image acquisition. In our implementation, we use WhisperX [25] as the speech transcription model, which provides highly accurate per-word timestamps.



scene_a1:00_extrinsics
Timestamp: 1733410104.084s
Relative offset: 16.063s
Frame 00241

Capture!

Figure 6: The user interface shows an operator recording calibration images in a synchronized multi-camera setup involving the DARKO robot platform (lower left). The operator prompts image acquisition during steady poses via speech commands, while having their hands free to securely hold and move the calibration target. The recognized spoken trigger word is shown on the right in yellow.



```
13:38:03 Initializing with method  "Vanishing Points" ...
13:38:03 Calibration initialized.
          status:  "Successful."

13:38:07 Calibration optimized.
          converged:  true
          rpe rms:  0.2614 px
          status:  "Successful."
```

Figure 7: Successfully converging calibration using an existing calibration software [23] and motion blur-free calibration images extracted using our proposed speech-guided calibration image acquisition approach.

3 T2.1: Object-Level Semantics

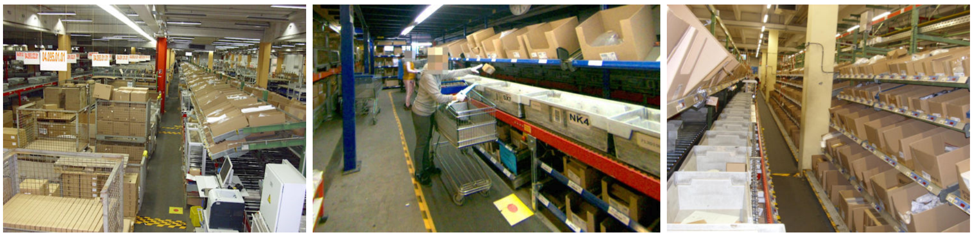


Figure 8: A typical warehouse environment in DARKO's lead use-case with uncommon object types, cluttered and dynamic scenes illustrating the challenges for object-level semantics.

The goal of task T2.1 is to develop a module providing real-time semantic and spatial information of the surroundings using robot's on-board sensors while considering the DARKO objective O3 on efficient deployment. This goal encompasses several challenges and research questions in the context of DARKO's target scenario (Figure 8): i) How to reduce the manual labeling effort to enable efficient deployment in environments underrepresented in large-scale public data sets?, ii) How to leverage multi-modal and multi-view data from the robot's sensors to obtain more robust semantics?, iii) Which deep neural network architecture to use under data and hardware constraints present in DARKO?, iv) Which semantic representations are useful for the downstream robot tasks?

We reported the interim progress towards answering these questions in the ***deliverable D2.2 Initial Report on Perception in DARKO***. In the following sections we discuss the developments made in the final phase of the project since the last report. For convenience we briefly summarize the key contributions reported in ***D2.2*** in addition to the ones described in this deliverable.

Key contributions in T2.1 (Reported in D2.2)

- Several data recording campaigns in a warehouse from our target use-case and in the Bosch robotics lab with a replica of the DARKO scenario, using a hand-held sensor suite resembling the DARKO robot's sensor setup.
- Development of an efficient 3D labeling strategy for static objects using SLAM to enable training of 3D object detectors with reduced manual annotation effort.
- Creation of an annotated 3D intralogistics dataset using the developed efficient labeling strategy for training and benchmarking of 9DoF 3D object detectors in scenarios relevant to DARKO use-cases.
- Extension of the RGB-D YOLO [26] deep neural network to support estimation of multiple keypoints, multiple object classes, 9DoF object bounding box estimation and RGB + 3D lidar input data.
- Deployment and integration of the object-level semantics module for real-time inference on the DARKO robot.
- Extension of modern 3D object detectors for 9DoF object detection including their training and evaluation on the DARKO intralogistics dataset with 18 object classes.
- Research on self-supervised learning for label-efficient 3D object detection, and for prediction of 3D scene graphs – a higher-level scene representation which models relationships between objects [7, 8, 9, 10].

- Research on alternative approaches for object-level semantics, such as a 2D-to-3D uplifting approach, which can potentially generalize better to a fisheye+lidar setup.
- Research on leveraging mid-level visual representations for more efficient semantics-aware navigation in collaboration with WP6 [11, 12].

Key contributions in T2.1 (since D2.2)

Our contributions achieved for task T2.1 in the last phase of the project and discussed in this deliverable include:

- **DARKO Intralogistics Dataset Enhancement:** Transferred annotations to Azure Kinect RGB-D data from the RGB + lidar DARKO intralogistics dataset using our auto-labeling pipeline (see D2.2 for details) to create a corresponding dataset version for cross-sensor generalization studies.
- **Cross-Sensor Generalization Analysis:** Systematically evaluated RGB-only, RGB-D, and point cloud-based 3D object detection methods for robustness to sensor changes using the enhanced dataset.
- **TR3D Adaptation and Integration for 9DoF Detection:** Extended the real-time capable TR3D method [20] for 9DoF object detection and trained it on the DARKO dataset. A ROS node was implemented, adhering to the DARKO object-level semantics module interface, and integrated into the DARKO perception stack for robot deployment.
- **360° Perception Pipeline Development:** Leveraged the adapted TR3D to develop a full surround object detection pipeline, effectively utilizing the robot's fisheye camera setup for comprehensive 360° perception.
- **Exploration of Open-Vocabulary Object-Level Semantics:** Assessed the potential of emerging techniques for open-vocabulary understanding of intralogistics scenes, including 3D Gaussian Splatting-based methods (e.g., OpenSplat3D [27]) on the DARKO dataset and approaches leveraging 2D foundation models (e.g., Grounded SAM [28, 29, 30]) with robot-captured data.
- **Advancements in Open-Vocabulary 3D Scene Understanding and Relational Modeling:** Conducted research and developed novel approaches for richer semantic understanding of 3D scenes, including methods for predicting open-vocabulary 3D scene graphs and for modeling object relationships within neural radiance fields, resulting in two publications [1, 2].
- **Scene Graph Applications for Embodied AI (in collaboration with WP6):** Investigated the use of 3D scene graphs and VLMs for open-vocabulary embodied tasks, exemplified by the GraphEQA [3] framework.

3.1 Advancements in 9DoF Object Detection for Intralogistics Environments

A primary goal for the DARKO robot's perception system was to achieve comprehensive 360° awareness of its surroundings, enabling robust navigation and interaction in complex intralogistics environments. The target sensor setup for this comprised two fisheye cameras (front and rear) and a 3D lidar. This section details the iterative development process, starting from addressing initial sensor domain gap challenges to the final deployment of a 360° 9DoF object detection pipeline based on an adapted TR3D method.

3.1.1 Investigating Sensor Domain Gaps and Selecting a Robust Baseline

The DARKO intralogistics dataset was primarily recorded using a hand-held sensor suite equipped with RGB cameras and a 3D lidar, reflecting the multi-modal input desired for robust perception. However, to rapidly prototype and deploy an initial object-level semantics module for T2.1, we first utilized an Azure Kinect RGB-D camera on the robot. The initial object detection model, based on RGB-D YOLO++ (as reported in D2.2), was trained primarily on RGB+Lidar data from our dataset. When this model was deployed using the Azure Kinect, a noticeable performance degradation occurred due to the sensor domain gap. This highlighted the critical challenge of ensuring perception models generalize well across different sensor modalities and hardware.

Given that the DARKO robot's final sensor configuration (2x fisheye cameras + 3D lidar) would again differ from both the original dataset's RGB cameras and the interim Azure Kinect setup, a systematic investigation into cross-sensor generalization became of interest. We enhanced the DARKO intralogistics dataset by transferring existing 9DoF object annotations from the RGB+Lidar stream to the simultaneously recorded Azure Kinect RGB-D data, using our auto-labeling pipeline (detailed in D2.2). This created a corresponding Azure Kinect version of the dataset. We provide statistics of the validation split used in the generalization experiments in the Figure 9. Further information about the objects and categories can be found in the deliverable D2.2.

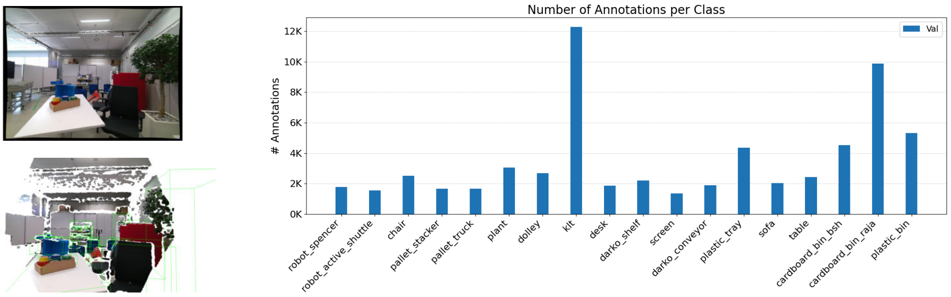


Figure 9: Data sample from Azure Kinect dataset with transferred labels on the left and per-class annotations statistics of the validation split used for cross-sensor generalization experiments.

Using this enhanced dataset, we performed comprehensive experiments evaluating various 9DoF 3D object detection methods benchmarked in D2.2, including RGB-only, RGB-D, and point cloud-only approaches. The objective was to quantify how models trained on one sensor configuration (e.g., portable sensor box RGB+Lidar) perform when inferring on another (e.g., Azure Kinect RGB-D). As illustrated in Figure 10, these experiments confirmed our initial findings from D2.2: point cloud-based methods (VoteNet-9DoF [31]) and RGB+Point Cloud fusion methods (ImVotenet-9DoF [32], DeMFVotenet-9DoF [33]) demonstrated significantly better resilience to changes in sensor hardware compared to methods relying heavily on RGB (Cube R-CNN [34]) or specific RGB-D characteristics (RGB-D YOLO++). Notably, DeMFVotenet-9DoF achieved the highest mAP@25 and mAP@50 scores, indicating superior performance in both detecting objects and accurately localizing them under these cross-sensor conditions. Conversely, RGB-D YOLO++ and Cube R-CNN exhibited a substantial drop in performance, underscoring their sensitivity to the domain shift. While additional augmentations for RGB/RGB-D methods could have potentially reduced overfitting, the strong generalization of point cloud-based approaches seemed to be a more promising direction.

This conclusion steered our focus towards point cloud-based methods for the final object-level semantics module. We sought a method that was not only accurate and robust

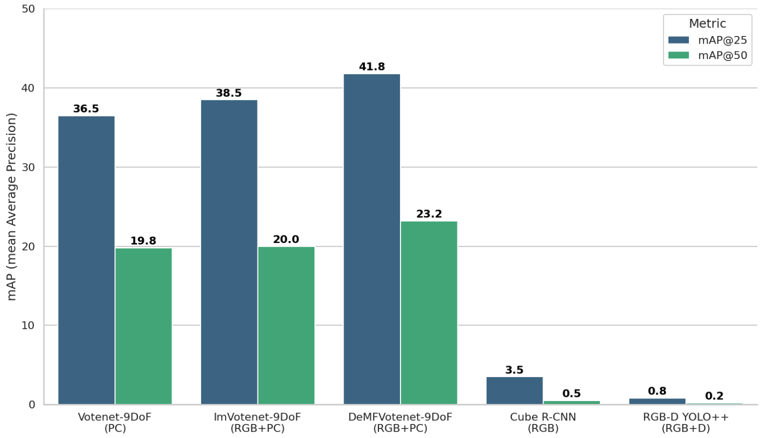


Figure 10: Cross-sensor generalization performance of various 9DoF 3D object detection methods on the DARKO intralogistics dataset. Models were trained on data from the portable sensor box (RGB+Lidar) and tested on data from the Azure Kinect (RGB-D). The plot shows mAP metrics at 0.25 (blue) and at 0.50 (orange) IoU levels.

to sensor variations but also real-time capable for deployment on the robot's hardware. This led us to investigate the TR3D method [20], a fully-convolutional 3D object detection model that demonstrated strong performance on standard indoor benchmarks like SUN RGB-D [35] while maintaining high inference speed.

3.1.2 9DoF TR3D on the DARKO Intralogistics Dataset

Having identified TR3D as a promising candidate due to its performance on point cloud data and its real-time capabilities, the next step was to adapt it for the specific requirements of the DARKO project and evaluate its efficacy on our challenging intralogistics dataset. The original TR3D method primarily focused on 7DoF object detection (3D position, 3D extents, and yaw).

Our key adaptations and contributions were:

- **Extension to 9DoF:** We modified the prediction head of the TR3D network to regress the full 9 degrees of freedom for object bounding boxes, including 3D centroid, 3D orientation (roll, pitch, and yaw), and 3D extents. This aligned the output with the established semantic representation used across DARKO tasks.
- **Training on DARKO Intralogistics Dataset:** The extended 9DoF TR3D model was trained trained on the DARKO intralogistics dataset with extensive hyperparameter tuning.
- **Data Augmentation Strategies:** To further improve performance and robustness, we investigated and incorporated various data augmentation techniques during training. These included global color augmentations for the point cloud, random point cloud sub-sampling (choosing a fraction of points randomly), and standard geometric augmentations such as random translation, scaling, and rotation of scenes.

The adapted 9DoF TR3D model demonstrated state-of-the-art performance on the DARKO intralogistics dataset compared to previously evaluated methods. Figure 11 presents the final benchmark results, where it achieves 63.2 mAP@25 and 43.4 mAP@50 on the validation set. This marks a substantial improvement over other strong baselines, including the best-performing RGB+D method, Cube R-CNN (54.0 mAP@25), and the best previous RGB+Point Cloud fusion method, DeMFVotenet (50.0 mAP@25).

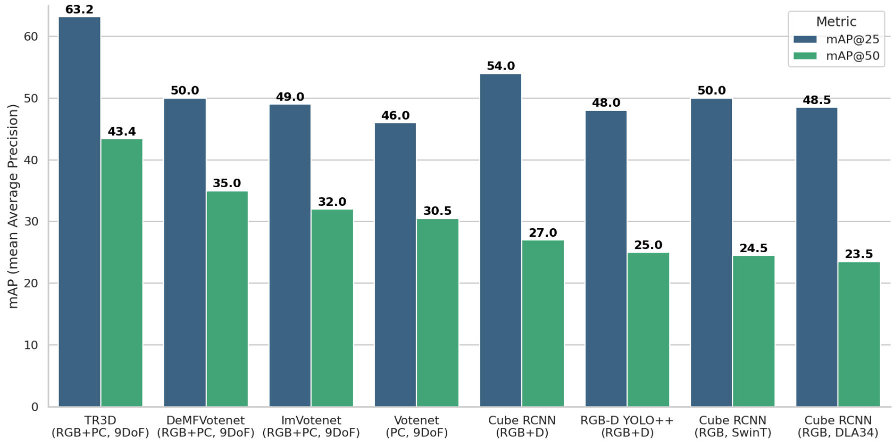


Figure 11: Final benchmark results on the validation split of the DARKO intralogistics dataset. Our adapted TR3D (RGB+PC, 9DoF) model significantly outperforms other baseline methods across both mAP@25 and mAP@50 metrics.

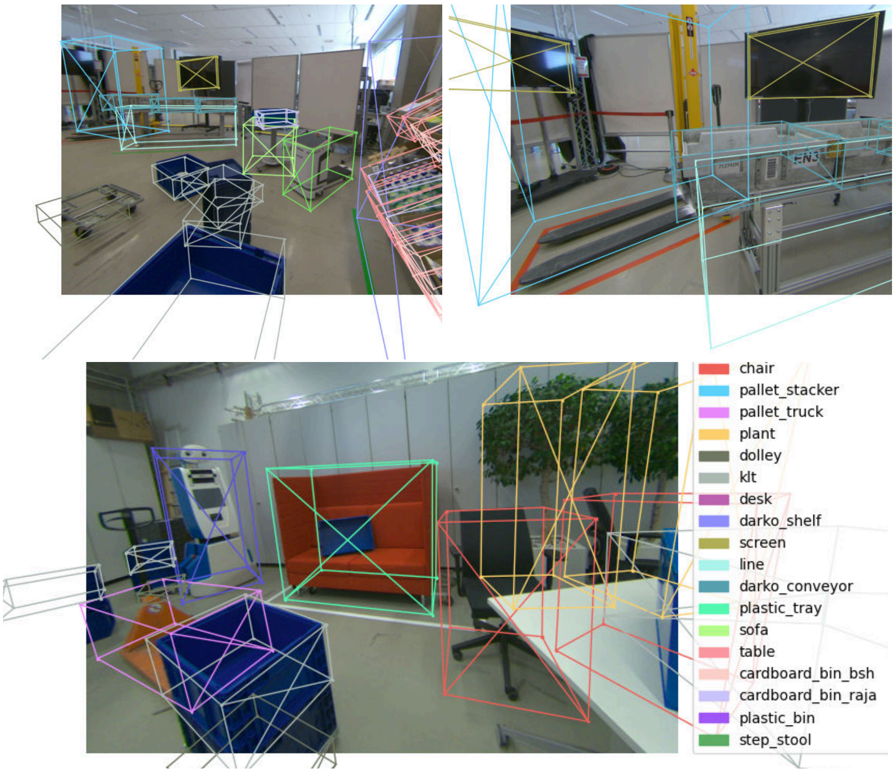


Figure 12: A few prediction examples on the validation scenes by our extended TR3D method. The *x* in the visualization indicates the front of the object.

Figure 12 shows a few qualitative examples of our extended TR3D model. In general, the method can handle quite well the objects extending out of the field of view and has a good recall and localization accuracy. It struggles a bit with stacked objects (e.g. KLTs) or objects rotated significantly with respect to the sensor. The former issue is likely caused by NMS, while the latter can potentially be addressed with stronger rotational augmentations.

In addition to its strong performance in the standard evaluation, we also assessed the cross-sensor transfer capability of our final TR3D model. When trained on the original RGB+Lidar data and tested on the Azure Kinect data, the model achieved the results shown in Table 1 (in addition to average precision metrics we also provide average recall AR @25 and @50). As with the previously evaluated methods, performance in the cross-sensor generalization task drops when compared to the results in Figure 11, but the model still maintains a reasonable detection capability (e.g., 45.83 AP@25). While the AP@50 score of 20.74 is slightly lower than the 23.2 achieved by DeMFVotenet-9DoF (see Figure 10), our model achieves a significantly higher AP@25 score (45.83 vs. 41.8). This indicates that TR3D is better at correctly detecting objects under a domain shift, even if the localization is slightly less precise for the stricter IoU threshold. Considering its state-of-the-art performance on the main benchmark (Figure 11) and its superior detection recall (AP@25) in the cross-sensor scenario, these results further validate our choice of TR3D as the most robust and well-rounded model for the project's hardware and operational requirements.

Table 1: Cross-sensor transfer performance of the adapted 9DoF TR3D model, trained on RGB+Lidar data and tested on Azure Kinect data.

	AP@0.25	**AR@0.25**	**AP@0.50**	**AR@0.50**
TR3D (RGB+PC, 9DoF)	45.83	71.63	20.74	31.93

It is worth noting that while TR3D can be augmented with 2D feature extractors from RGB images for higher localization accuracy, common pre-trained feature extractors are not directly applicable to the heavily distorted fisheye camera images without significant adaptation or re-training. Furthermore, incorporating such feature extractors would likely increase computational requirements and reduce runtime performance. Therefore, for the deployed system, we opted to work directly with the colored point clouds balancing performance and accuracy for real-world robotic operation.

3.1.3 Deployment of 360° TR3D Perception with Fisheye Cameras on the DARKO Robot

The final challenge was to deploy the adapted 9DoF TR3D model on the DARKO robot, leveraging its full 360° sensor suite consisting of front and rear fisheye cameras and a 3D lidar to achieve comprehensive surround perception.

This involved several key developments:

- **Multi-View RGB Point Cloud Projection Module:** A crucial component developed was a ROS C++ module for projecting color information from multiple RGB cameras onto the 3D lidar point cloud. This module supports various camera models, including pinhole, fisheye, and the double-sphere, which was specifically used for the robot's front and rear fisheye cameras. The module synchronizes the input data streams and generates a dense, 360° field-of-view colored point cloud.
- **TR3D ROS Node Implementation:** A dedicated ROS node was implemented to perform inference using the adapted 9DoF TR3D model on the 360° colored point clouds generated by the projection module.

- **Addressing Limited Field-of-View in Training Data:** Our DARKO intralogistics dataset, while comprehensive, primarily captured scenes with a frontal field of view (approximately 120°). Directly running the TR3D model trained on this data on the full 360° point cloud led to degraded detection performance, particularly for objects in the rear field of view. To mitigate this, we implemented a strategy where the input 360° point cloud is split into two overlapping segments: a frontal view and a rear view (each covering approximately 220° to ensure overlap). The rear point cloud segment is rotated by 180° to align its principal viewing direction with that of the frontal training data before being fed to the TR3D model for inference with a batch size of 2. Predictions from this "virtual frontal" view of the rear segment are then rotated back to their original orientation.
- **Non-Maximum Suppression (NMS) for Fused Detections:** After obtaining detections from both the front and (rotated) rear point cloud segments, a final 3D Non-Maximum Suppression (NMS) step is applied. This effectively removes redundant or overlapping detections, particularly in the regions where the front and rear fields of view overlap, yielding a consistent set of 360° object detections.

To determine the optimal configuration for deployment, we investigated different parametrizations of our adapted TR3D model, focusing on the trade-offs between accuracy, inference speed, and resource usage. Table 2 summarizes these findings. We evaluated two different backbones, TR3DMinkResNet34 and a lighter TR3DMinkResNet18, as well as various voxel sizes for the input point cloud. The results show a clear relationship between voxel size and performance: while a smaller voxel size of 0.5cm yields a slight improvement in mAP@50, it comes at a significant cost to inference speed. Conversely, larger voxel sizes drastically reduce accuracy. The GPU memory footprint for all tested variants was well within the limits of the robot's hardware. Ultimately, we selected the TR3DMinkResNet34 backbone with a 1cm voxel size, as it offered the best trade-off, achieving our highest mAP@25 of 63.2 while maintaining a fast inference time. The inference speed for a batch size of 2 is particularly relevant, as this reflects the parallel processing of the front and rear point cloud segments in our deployed 360° pipeline.

Table 2: Performance and resource usage of different TR3D variants on RTX 3060 GPU.

Backbone	**Voxel size (cm)**	**Params (M)**	**GPU (GB)**		**Inference (s)**		**mAP**	
			BS 1	**BS 2**	**BS 1**	**BS 2**	**@50**	**@25**
TR3DMinkResNet34	0.5	14.70	0.57	1.08	0.21	0.39	44.6	62.9
	1.0		0.22	0.39	0.10	0.16	43.4	63.2
	2.0		0.12	0.18	0.06	0.10	26.8	51.0
	5.0		0.09	0.11	0.05	0.07	19.2	5.0
TR3DMinkResNet18	1.0	2.10	0.12	0.22	0.07	0.12	37.7	59.3

This deployed pipeline, integrated into the DARKO perception stack, adheres to the defined DARKO interfaces for seamless communication with downstream modules e.g. it provides information for manipulation and navigation components about the locations of the shelf, the storage bins, conveyor with plastic trays, etc. We have also integrated the object-level semantics module with human perception (detailed in the description of T2.5) for pointing gesture recognition, which can be used e.g. for target tray or source bin selection. On the DARKO robot's hardware, equipped with an NVIDIA RTX 3060 GPU, the 360° 9DoF TR3D perception system achieves an inference rate of approximately 5 Hz with a latency of around 0.5 seconds. Qualitative examples of the 360° perception on the robot in two different environments are shown in Figures 13 and 14.

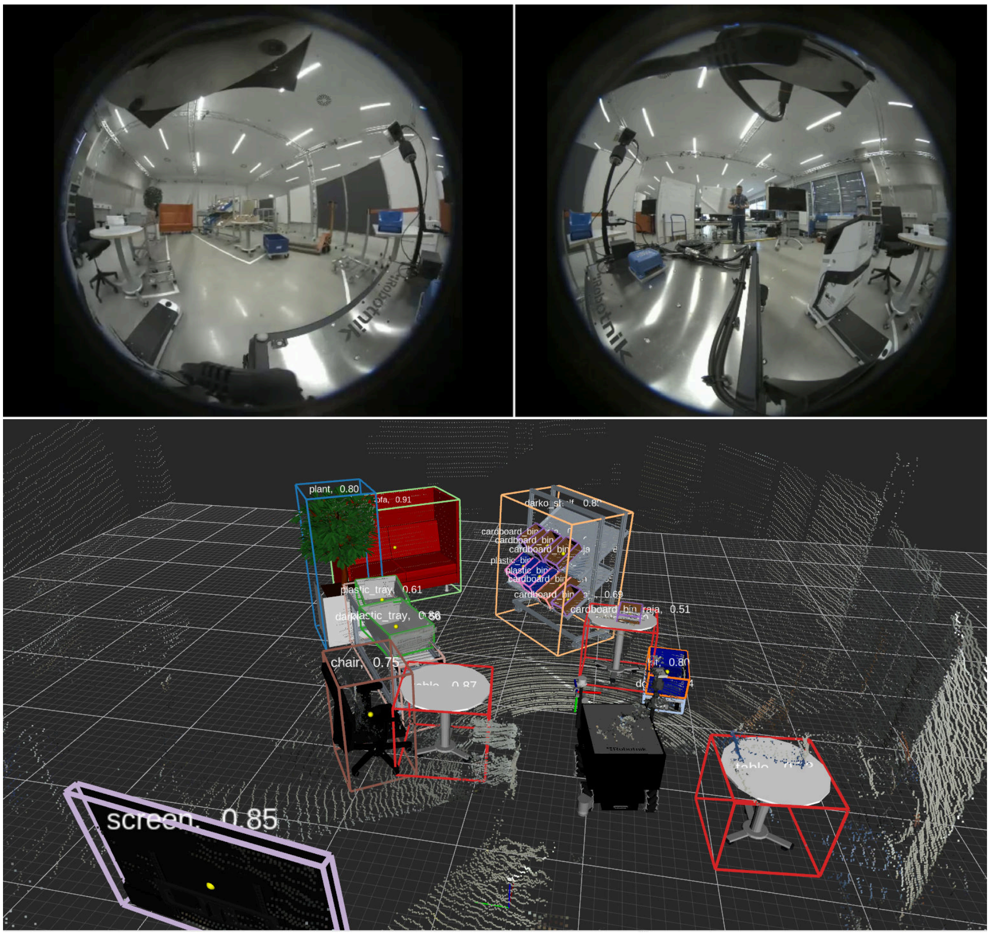
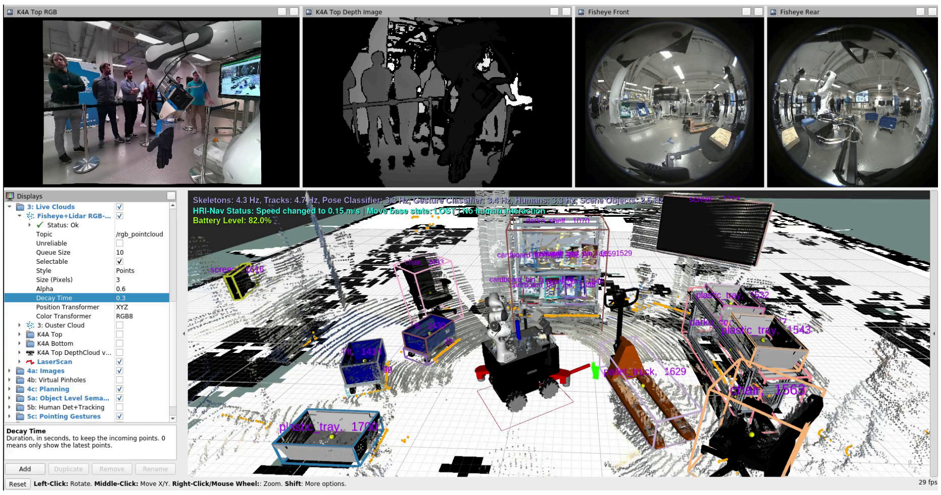


Figure 13: Top: Views from the front and the rear fisheye cameras mounted on the DARKO robot. **Bottom:** Single frame 9DoF 3D object detections generated by the deployed pipeline utilizing TR3D. The object meshes are scaled and oriented based on the estimated extents and poses. Note that objects are detected behind and in front of the robot.

3.2 Exploration of Open-Vocabulary Object-Level Semantics

While robust detection of known object categories is essential, the ability of a robot to understand and interact with novel, previously unseen objects and to comprehend complex scenes in a more human-like, open-ended manner is a key objective for advanced robotics. This section details our exploration into open-vocabulary perception, moving beyond fixed-category detection towards more flexible scene understanding.

In D2.2, we reported initial promising results from applying the Segment Anything Model (SAM) [29] to data from the DARKO robot, noting its promising generalization to fisheye camera images. To extend this capability from class-agnostic segmentation to open-vocabulary object detection, we investigated Grounded SAM [28, 30], which combines SAM with a text-promptable object detector. We applied Grounded SAM to fisheye images recorded on the DARKO robot at the ARENA2036 facility in Stuttgart. Figure 15 shows qualitative examples where the model was prompted with various DARKO-related object categories. While the results sometimes demonstrate strong zero-shot performance, they also reveal several key limitations. The method struggles to consistently identify smaller or more domain-specific objects, such as individual bins on a shelf or DARKO conveyor.



3: Live Clouds ☑
Fisheye+Lidar RGB-... ☑
✓ Status: Ok
Topic /rgb_pointcloud
Unreliable □
Queue Size 10
Selectable ☑
Style Points
Size (Pixels) 3
Alpha 0.6
Decay Time 0.3
Position Transformer XYZ
Color Transformer RGB8
3: Ouster Cloud □
K4A Top □
K4A Bottom □
K4A Top DepthCloud v... □
LaserScan ☑
4a: Images ☑
4b: Virtual Pinholes □
4c: Planning ☑
5a: Object Level Sema... ☑
5b: Human Det+Tracking □
5c: Pointing Gestures ☑

Decay Time
Duration, in seconds, to keep the incoming points. 0 means only show the latest points.

Skeletons: 4.3 Hz, Tracks: 4.7 Hz, Pose Classifier: 3.3 Hz, Gesture Classifier: 3.4 Hz, Humans: 3.3 Hz, Scene Objects: 2.5 Hz
HRI-Nav Status: Speed changed to 0.15 m/s | Move base state: LOST | No human interaction
Battery Level: 82.0%

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

Figure 14: Live view from DARKO robot in KI.Fabrik @ Deutches Museum during the final integration week with surround-view 9DoF 3D object detections. Only fisheye cameras and 3D lidar were used to generate object-level semantics shown in the bottom part of the figure. The pipeline is running with all the other DARKO components enabled. The detected objects include specific darko objects like *darko_shelf*, *plastic_tray*, *darko_conveyor*, and other intralogistics items such as storage bins, KLTs, and a pallet truck.

Furthermore, we observed a number of false-positive detections and a general lack of consistency across different views. These inconsistencies highlight the need to enforce multi-view geometric constraints. Also, these foundation models operate in 2D and require a robust mechanism to "uplift" their understanding into a coherent 3D representation. This motivated us to look into methods that are inherently 3D and can naturally integrate information from multiple views. We did this in alignment and collaboration with DARKO WP3.

To address the challenges of 3D uplifting and multi-view consistency, we explored methods based on 3D Gaussian Splatting (3DGS) [36], a technique that creates an explicit, differentiable 3D representation of a scene from a collection of images. 3DGS provides a natural framework for distilling features from 2D foundation models into a consistent 3D representation. For this purpose, we employed the OpenSplat3D [27], which learns instance-level features on the 3D Gaussians supervised by 2D instance masks from SAM. We applied this approach to sequences from our DARKO intralogistics dataset. Our qualitative analysis with some examples shown in Figure 16 indicates that this method can successfully segment common, well-defined objects like chairs, sofas, and tables. However, we also observed a persistent domain gap when dealing with intralogistics-specific objects. For instance, the method often struggled with the DARKO shelf, bins, and conveyor trays, resulting in under- or over-segmentation. This is evident in cases like failing to separate stacked KLTs, incorrectly segmenting trays on the conveyor, or splitting single objects like pallet truck/stacker into multiple parts.

Our investigation into open-vocabulary methods reveals both significant potential and practical limitations for real-time robotics. The primary challenges include:

- **Computational demands:** Methods based on 3DGS, like OpenSplat3D, are computationally intensive and require significant offline processing time (e.g., 20-45 minutes per scene), making them unsuitable for real-time operation on a mobile robot.

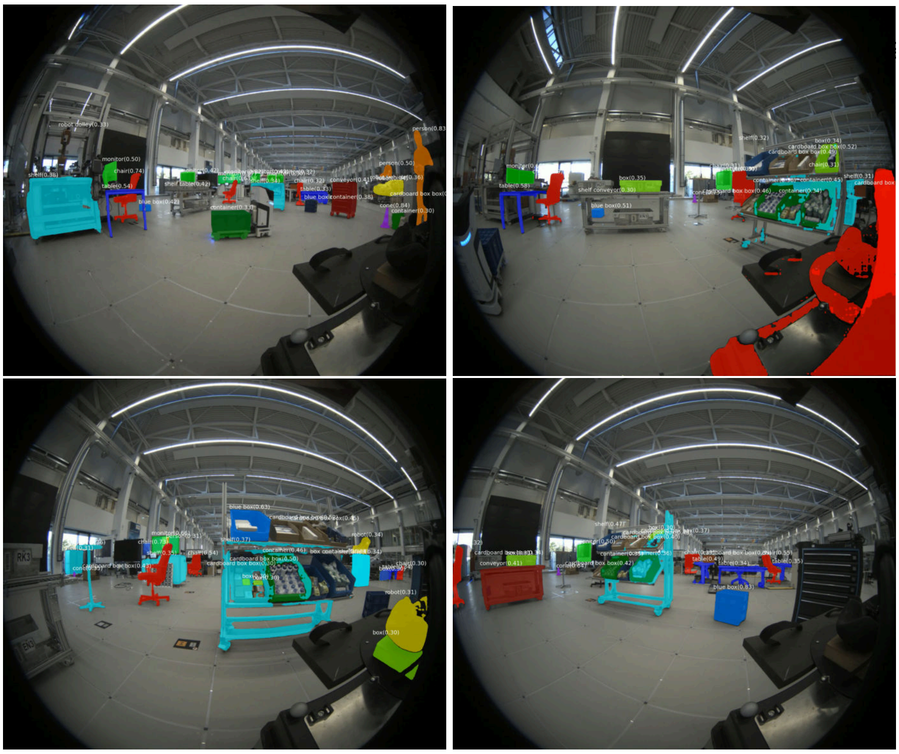


Figure 15: Qualitative examples of Grounded SAM on fisheye images recorded at ARENA2036 in Stuttgart. Different colors correspond to different catageries (e.g. *shelf*, *blue box*, *box*, *person*, *monitor*, *conveyor*, *container*, *table*, *chair*, etc.) used for prompting.

- **Static scene assumption:** These reconstruction-based methods fundamentally assume a static scene during data capture, which is a major constraint in dynamic intralogistics environments.
- **Domain gap:** Despite their impressive generalization, we observed a clear domain gap. Foundation models trained on general web-scale data still struggle with the unique appearance and context of specialized industrial objects, leading to inconsistent or incorrect segmentation.

Nevertheless, these methods hold considerable promise for offline applications within the DARKO scope. Their ability to perform zero-shot segmentation can be leveraged for highly efficient data annotation. By reconstructing a scene and performing open-vocabulary segmentation, one can generate vast amounts of 3D labels for training more lightweight, real-time detectors, directly contributing to the DARKO objective of reducing manual labeling effort and enabling more efficient deployment in novel environments.

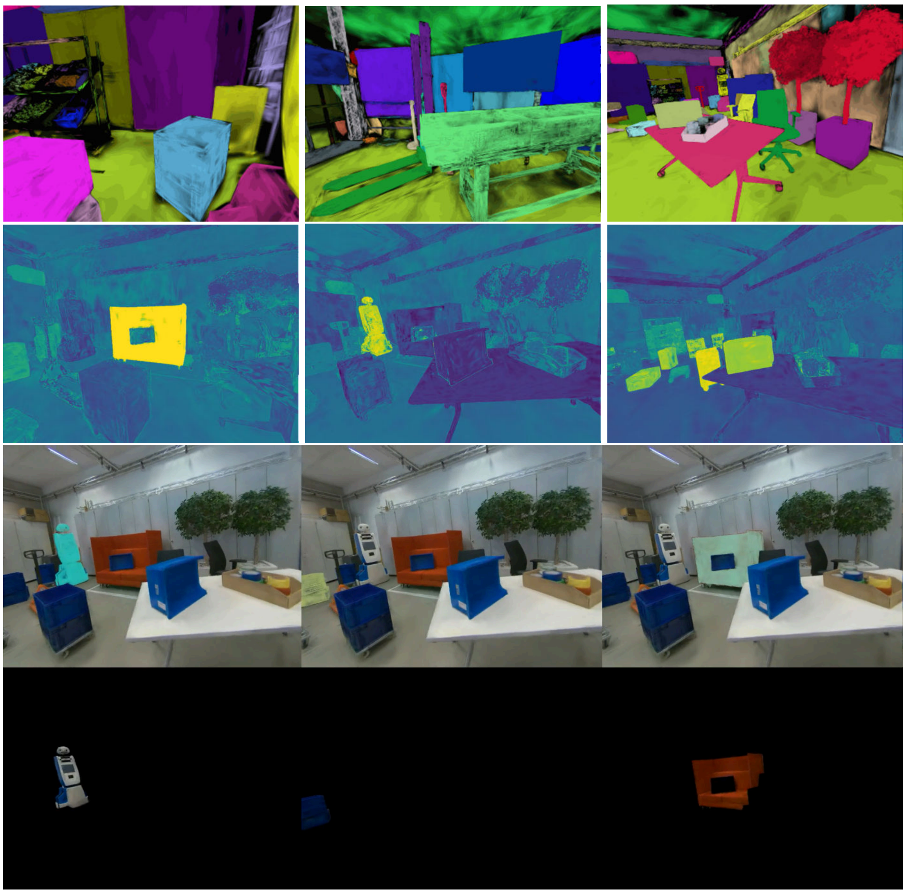


Figure 16: Example results of OpenSplat3D on one of the sequences from DARKO intralogistics dataset. **Top:** instance segmentation with different color representing different instances (black color corresponds to rendering of unassigned/noisy Gaussians). **Middle:** Gaussian field response to language queries *sofa*, *robot*, *blue box* respectively from the left to the right. **Bottom:** occlusion-aware RGB renderings of different instances, highlighting the method's ability to generate 2D/3D mask labels.

3.3 Open-Vocabulary 3D Scene Understanding and Relational Modeling

As discussed in D2.2, higher-level scene representations that move beyond object-centric semantics to encode the relationships between objects are highly relevant for DARKO. This is a research direction that Bosch pursues, and it has been partially funded by the DARKO project[1]. Our recent work has focused on pushing these capabilities into the open-vocabulary domain, where neither the objects nor the relationships are from a predefined list. This research has led to two publications at CVPR, namely Open3DSG [1] and RelationField [2], which we briefly summarize below.

Open3DSG introduces a method to predict a 3D scene graph directly from a point cloud in an open-vocabulary fashion without requiring labeled scene graph data. The approach (Figure 17) co-embeds the features from a 3D scene graph prediction backbone with the feature space of powerful open world 2D vision language foundation models. This enables Open3DSG to predict 3D scene graphs from 3D point clouds in a zero-shot manner by

[1] DARKO funds one of the co-supervisors of the PhD student working on this topic.

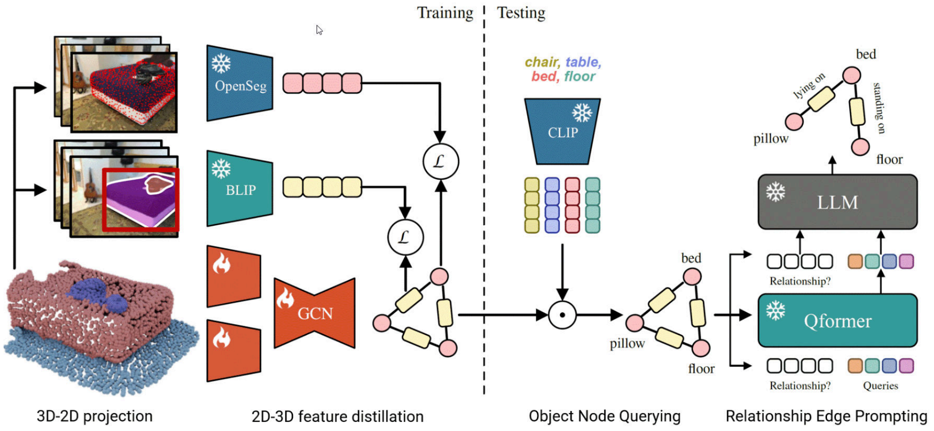


Figure 17: Open3DSG overview. Given a point cloud and RGB-D images with their poses, Open3DSG distills the knowledge of two vision-language models into a GNN. The nodes are supervised by the embedding of OpenSeg [37] and the edges are supervised by the embedding of the InstructBLIP [38] vision encoder. At inference time, Open3DSG first computes the cosine similarity between object queries encoded by CLIP [39] and our distilled 3D node features to infer the object classes. Then it uses the edge embeddings as well as the inferred object classes to predict relationships for pairs of objects using a Qformer & LLM from InstructBLIP

querying object classes from an open vocabulary and predicting the inter-object relationships from a grounded LLM with scene graph features and queried object classes as context. Experiments show that Open3DSG is effective at predicting arbitrary object classes as well as their complex inter-object relationships describing spatial, supportive, semantic and comparative relationships. Open3DSG can be queried and prompted depending on the current task context, this is especially interesting for robotics where required semantic information often depends on the downstream application. Some examples of predicted scene graphs and additional semantic information are show in Figure 18. Please refer to Open3DSG [1] for more details.

RelationField is the first method to extract inter-object relationships directly from neural radiance fields. It represents relationships between objects as pairs of rays within a neural radiance field, effectively extending its formulation to include implicit relationship queries. RelationField learns complex, open-vocabulary relationships by knowledge distillation from multi-modal LLMs (see Figure 19). Experiments show that RelationField achieves state-of-the-art performance in open-vocabulary 3D scene graph generation and relationship-guided instance segmentation tasks. A few qualitative examples in Figure 20. Please refer to [2] for more details.

We have also tried Open3DSG and RelationField on DARKO intralogistics dataset. Our initial findings suggest that both methods do not generalize well to ingralogistics use-case, potentially due to aforementioned domain gap, but deeper analysis is needed to fully understand the potential of these methods in industrial environments. Together with WP3, We are also investigating in an going master thesis hierarchical 3D Scene Graphs, which can be build incrementally and represent larger environments beyond single scenes. These results will be reported in the final periodic report.

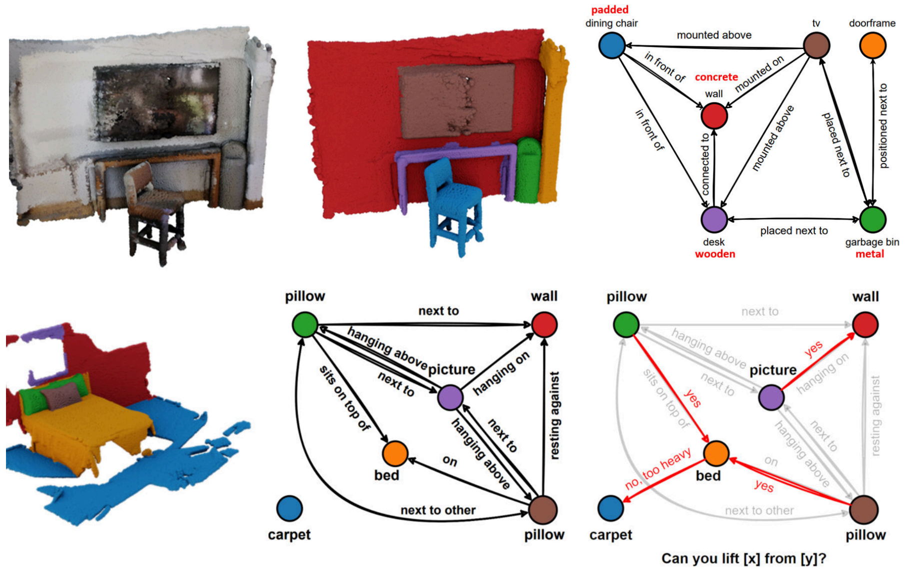


Figure 18: Example predicted scene graphs by Open3DSG in indoor scenes. Additional atrtibutes and inter-object affordances can be retrieved by prompting the LLM. This additional semantic information is marked in red.

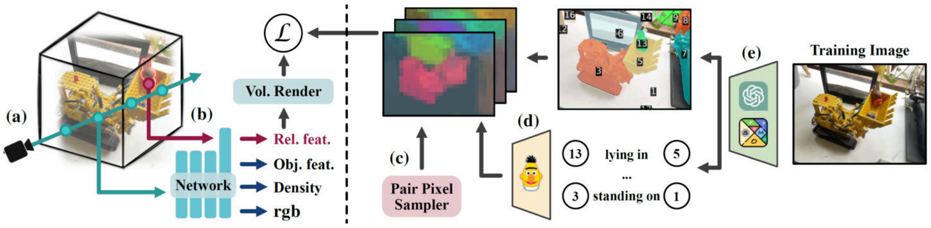


Figure 19: Left: RelationField learns a 3D feature field (a) that can be queried with a relationship query location (b) which changes the relationship field of the 3D volume depending on what position is selected. The relationship feature is sampled and rendered along a ray according to NeRF's rendering weights. The language loss maximizes the cosine similarity between the extracted sparse features from the 2D views and the rendered 3D relationship features. **Right:** 2D relationship proposals are estimated from a multi-model LLM prompted with SoM [40] (e) for each training view and encode extracted textual relationship description into the image plane (d). A pair pixel sampler samples subject and object pixels (c) for which the relationship feature is distilled into the 3D volume.

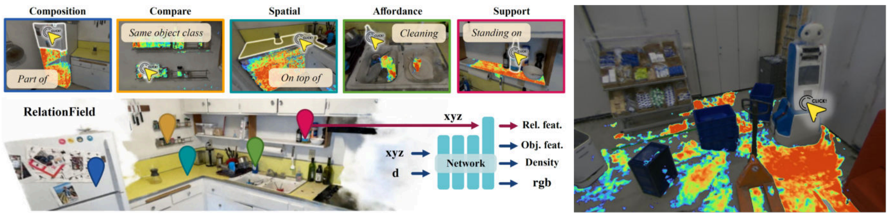
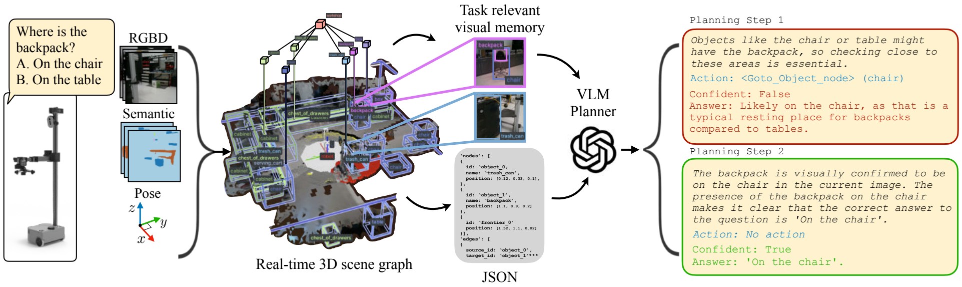


Figure 20: Qualitative examples of RelationField in a kitchen environment (left) and in DARKO intralogistics scene with "standing on" relationship query (right).



```
'nodes': [
{
  id: 'object_0',
  name: 'trash_can',
  position: [0.12, 0.33, 0.1],
},
{
  id: 'object_1',
  name: 'backpack',
  position: [1.1, 0.9, 0.2]
},
{
  id: 'frontier_0'
  position: [1.52, 1.1, 0.02]
}],
'edges': [
{
  source_id: 'object_0',
  target_id: 'object_1'•••
```

Planning Step 1

Objects like the chair or table might have the backpack, so checking close to these areas is essential.

Action: <Goto_Object_node> (chair)

Confident: False
Answer: Likely on the chair, as that is a typical resting place for backpacks compared to tables.

Planning Step 2

The backpack is visually confirmed to be on the chair in the current image. The presence of the backpack on the chair makes it clear that the correct answer to the question is 'On the chair'.

Action: No action

Confident: True
Answer: 'On the chair'.

Figure 21: GraphEQA is a novel approach for utilizing real-time 3D metric-semantic hierarchical scene graphs and task-relevant images as multimodal memory for grounding vision-language based planners to solve embodied question answering tasks in unseen environments.

3.4 Downstream Applications of 3D Scene Graphs

One of the research questions for T2.1 is to investigate which semantic representations are useful for downstream robotic tasks. To this end, and in collaboration with WP6, we explored the application of 3D scene graphs for complex tasks like Embodied Question Answering (EQA). EQA requires an agent to actively explore its environment to find answers to natural language questions, making it a great benchmark for the utility of a scene representation. The foundational work for this exploration was initiated during a PhD sabbatical funded by the DARKO project, and was subsequently completed as part of a larger academic collaboration at Carnegie Mellon University (CMU).

GraphEQA. The resulting framework, GraphEQA [3] (see als Figure 21), proposes a novel approach where an agent utilizes a real-time 3D metric-semantic scene graph (3DSG) and a curated set of task-relevant images as a multi-modal memory. As the robot navigates, it incrementally constructs the 3DSG. When a question is posed, the framework employs a hierarchical planner that leverages the structure of the scene graph for efficient, semantic-guided exploration. A Vision-Language Model (VLM), grounded by both the scene graph and the visual memory, acts as a high-level planner to generate common-sense exploration goals (e.g., first select a room, then an object within it). These goals are then translated into low-level, executable navigation commands. The work demonstrates that a dynamically built scene graph is a highly effective representation for such tasks, allowing the agent to significantly outperform baselines that rely on less structured memories by achieving a higher success rate with fewer planning steps.

In the context of DARKO use-case, this capability for interactive, semantic question answering is highly relevant. A robot equipped with this framework could answer complex queries from a human operator about the state of the warehouse, for example, "How many blue KLTs are on the second shelf of the rack in aisle five?". The GraphEQA framework was successfully demonstrated on a robot, showcasing its ability to answer complex questions in real world. In these experiments, the perception and some other heavier computations were handled by an external computer. While this validated the approach, a full integration with the onboard, real-time perception stack of the DARKO platform was not pursued due to resource and time constraints within the project. Nevertheless, this work serves as a demonstration of how the semantic representations such as 3D scene graphs can be utilized for sophisticated downstream tasks, with a clear path for future integration into robotic systems like DARKO.

4 T2.2: Perception for Manipulation

Task T2.2 deals with the challenge of determining suitable strategies for object picking using sensor observations, developing a module that provides stable and task-relevant grasps to inform object picking (T4.3).

This task addresses the core challenge of enabling reliable and efficient object picking in diverse and unstructured environments using sensor observations. The primary goal is to develop perception modules that can provide stable, task-relevant grasp proposals for downstream manipulation (T4.3), adapting to the wide variety of DARKO objects and scenarios encountered in real-world settings. Traditional approaches often rely on object-centric pipelines that first detect and segment objects, then assign pre-defined grasp points. However, these methods frequently struggle in cluttered or novel environments.

To overcome these limitations, T2.2 explores several complementary strategies. First, we introduce **VoteGrasp** [4], a novel method for directly regressing grasp poses from raw RGB-D or point cloud data, eliminating the need for explicit object detection. However, single-shot grasping is not always feasible, for example, with tightly packed items or flat objects lying on a table. To address such cases, we propose a hierarchical reinforcement learning approach for sequencing parameterized manipulation primitives (**ED-PMP** [13]), enabling the system to plan and execute multi-step manipulation strategies when a single grasp is insufficient. Furthermore, to tackle the challenge of data efficiency and sparse rewards, we introduce the **KEA** [5] approach, which combines novelty-based exploration with Soft Actor-Critic (SAC) to improve learning efficiency. Finally, as our next step, we investigate the use of large-scale pretraining via vision-language-action (**VLA**) models to further enhance generalization and reduce the amount of task-specific data required.

Key contributions in T2.2:

- **VoteGrasp**: a novel method for grasp pose estimation, directly regressing grasp poses from RGB-D sensor data [4]. *Reported in D2.2.*
- Validation of generalising grasp poses generated by VoteGrasp, trained with a two-finger gripper, to grasping DARKO-specific objects (including objects in transparent plastic bags) with the five-fingered SoftHand – without re-training. *Reported in D2.2.*
- **Extrinsic dexterity with parameterised motion primitives (ED-PMP)**: employing hierarchical reinforcement learning, including learning control policies for parameterised sub-tasks to improve training efficiency, for picking up objects with "occluded grasps" with the help of external surfaces. *Reported in D2.2 [14].* After D2.2, the final version of this work has been published at ICRA 2024 [13].
- **Deployment and integration** of VoteGrasp on the DARKO robot platform, integrating with planning and control from WP4. We have experimentally demonstrated the VoteGrasp perception system with a DH-3 gripper at MS2 (ARENA2036, Stuttgart) and Automatica 2023 (*reported in D2.2*). We have subsequently demonstrated VoteGrasp with a five-finger SoftHand at MS3 and MS4 (KI Fabrik, Munich) and will show it at Automatica 2025.
- **KEA** approach to efficient reinforcement learning in sparse reward settings. To extend our previous work [13] to multiple complex manipulation primitives *without* having to manually define dense reward functions for each sub-task, we have proposed a method that combines novelty-based exploration with SAC and proactively

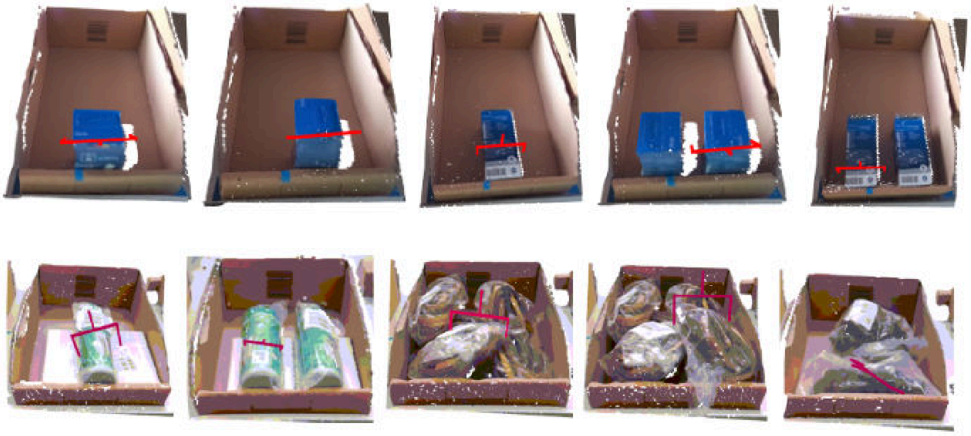


Figure 22: Examples of grasp pose estimations on DARKO objects, visualized as coloured point clouds. The top row shows objects in different configurations, while the bottom row presents the same objects wrapped in transparent plastic bags. Red markers indicate the selected grasp poses. In all cases, the grasp predictions are generated directly from point cloud data.

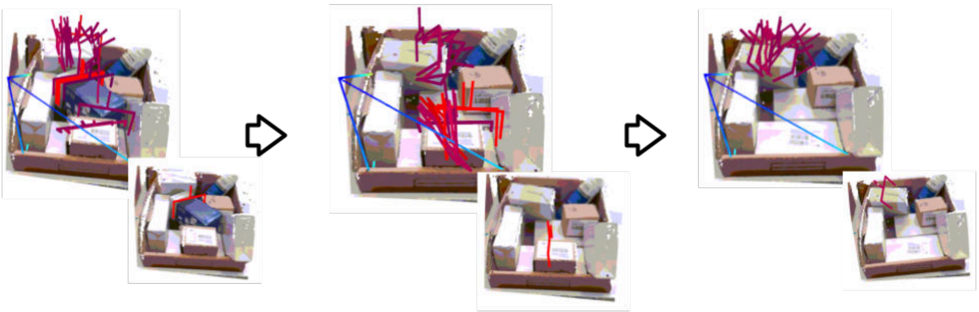


Figure 23: Examples of VoteGrasp grasp pose estimations in cluttered scenes, visualized from point cloud data. Multiple candidate grasp poses (shown in red and magenta) are generated for each step. These results show the model's ability to predict viable grasp configurations in moderately cluttered environments.

coordinates different exploration strategies, leading to more explicit learning with sparse rewards. This work will be published at ICML 2025 [5].

4.1 VoteGrasp: Grasp Pose Estimation in Cluttered Scenes

We proposed VoteGrasp, which is responsible for grasp pose estimation in cluttered scenes, was reported in D2.2. In summary, VoteGrasp is a perception system that directly regresses grasp poses from point cloud data, enabling robust object picking in complex and visually diverse environments. As described in D2.2, VoteGrasp was integrated into the DARKO robot platform and demonstrated with a DH-3 gripper at MS2 (ARENA2036, Stuttgart) and Automatica 2023. It was also applied with a five-finger SoftHand during the demonstrations at MS3 and MS4 (KI.FABRIK, Munich).

Our experiments show that VoteGrasp generalizes well to previously unseen objects, including challenging cases such as items enclosed in transparent plastic bags. Notably, the model is trained on a two-finger gripper dataset but performs effectively with a five-finger SoftHand without requiring any retraining. These results validate the first

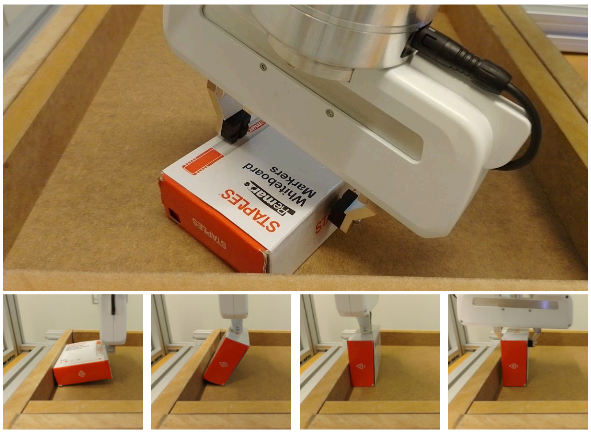


Figure 24: *Top:* In the initial pose, all feasible grasps on the target object are occluded by the environment. *Bottom (left to right):* ED-PMP learns to push the object to a wall and exploit it as a pivot to flip the object up and finally grasp it from the top.

interface envisioned in Task T2.2, where grasp proposals are generated for T4.3 to select suitable, collision-free grasps. Overall, this demonstrates the robustness of VoteGrasp and its suitability for real-world deployment.

To assess its generalization across object configurations, we further evaluated VoteGrasp on various DARKO objects arranged in different layouts, as shown in fig. 22 (top row). Additionally, we tested objects wrapped in transparent plastic bags (bottom row) with five-finger SoftHand, confirming the model's robustness to visual disturbances and its ability to generalize across different gripper types.

We also extended our evaluation to cluttered scenes, as shown in fig. 23. VoteGrasp performs reliably in moderately cluttered environments, but its performance degrades when objects are tightly packed. In such cases, direct single-shot grasping becomes infeasible, motivating the need for more advanced strategies such as manipulation primitives.

4.2 ED-PMP: Learning Primitives for Dexterous Manipulation

Certain flat or otherwise ungraspable objects cannot be picked up with a single top–down grasp; e.g., a book that is wider than the robot's gripper, lying on a table. Addressing such situations is central to DARKO's objective of task–level dexterous manipulation of everyday objects.

In Deliverable 2.2, we introduced ED-PMP [14], which learns a low-level primitive (*flipping*) with hierarchical reinforcement learning and lets a high-level policy sequence multiple primitives to complete the task. As reported in D2.2, ED-PMP focuses on improving grasping performance in challenging scenarios such as occluded or constrained environments. This method complements direct grasp pose estimation by providing an alternative strategy when collision-free grasps are not readily available. See fig. 24. We showed that learning primitives instead of hand-coding them yields robust and efficient grasps for previously inaccessible, flat objects.

This work has been refined and published as a full paper at ICRA 2024 [13], validating its effectiveness in improving training efficiency and grasp success in real-world tasks.

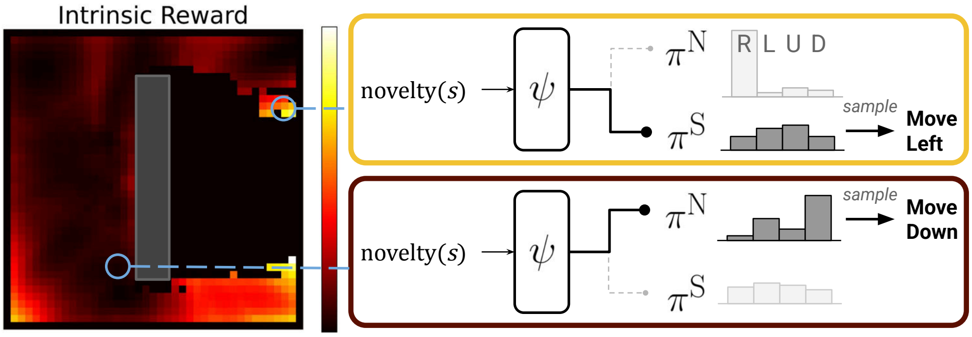


Figure 25: Overview. KEA integrates a novelty-augmented SAC ($\mathscr{A}^{\mathrm{N}}$) with a standard SAC agent ($\mathscr{A}^{\mathrm{S}}$). A switching mechanism (ψ) proactively coordinates between $\mathscr{A}^{\mathrm{N}}$ and $\mathscr{A}^{\mathrm{S}}$ based on the current state novelty computed by the novelty-based model. The stochastic policies, π^{N} and π^{S}, are derived from $\mathscr{A}^{\mathrm{N}}$ and $\mathscr{A}^{\mathrm{S}}$, respectively.

4.3 KEA: Improve Learning Efficiency under Sparse Rewards

A key limitation of the ED-PMP approach section 4.2 is that it relies on *task-specific dense reward functions* for each primitive to keep training efficient. Extending the framework to many new primitives quickly becomes infeasible because crafting reliable dense rewards demands expert knowledge and extensive tuning.

To remove the dense-reward bottleneck, we reformulate each primitive under a sparse reward setup. Agents receive a signal only upon successful completion. Sparse rewards reduce reward-engineering costs but make exploration significantly inefficient. We therefore developed KEA (Keeping Exploration Alive) [5], which addresses this challenge by augmenting Soft Actor–Critic (SAC) [41] with novelty-based intrinsic rewards [42, 43] to encourage exploration of unfamiliar states.

As shown in fig. 25, we integrate an additional "standard" agent (denoted as $\mathscr{A}^{\mathbf{S}}$) with the novelty-augmented agent (denoted as $\mathscr{A}^{\mathbf{N}}$), providing a complementary exploration strategy to address inefficiencies caused by the complexity of interactions between the exploration strategies. To coordinate $\mathscr{A}^{\mathrm{N}}$ and $\mathscr{A}^{\mathrm{S}}$, we devise a switching mechanism, denoted as ψ, which dynamically coordinates based on state novelty, measured by the novelty-based model.

Because SAC has demonstrated significant success in continuous control tasks, we use it as the base RL agent and leverage Random Network Distillation (RND) [42] to compute intrinsic reward for exploration ($\mathscr{A}^{\mathrm{N}}$). In an off-policy manner, we can collect transitions with multiple policies while training with another. This allows us to use distinct exploration strategies (e.g. $\mathscr{A}^{\mathrm{N}}$ and $\mathscr{A}^{\mathrm{S}}$) to gather diverse data from the environment.

We first test KEA in a 2-D navigation task, where the agent must navigate to a fixed goal position while avoiding obstacles. Then, we test KEA in DeepSea [44], a hard exploration Benchmark that consists of an $N \times N$ grid where the agent starts in the top-left corner and aims to reach a goal in the bottom-right cell. Finally, we evaluate KEA in continuous control tasks with a sparse reward setup in the DM Control Suite [45]. Results are shown in fig. 26 and tables 3 and 4. In multiple hard exploration tasks, KEA achieves comparable or better performance to the state-of-the-art methods.

Full details appear in our ICML 2025 paper [5].

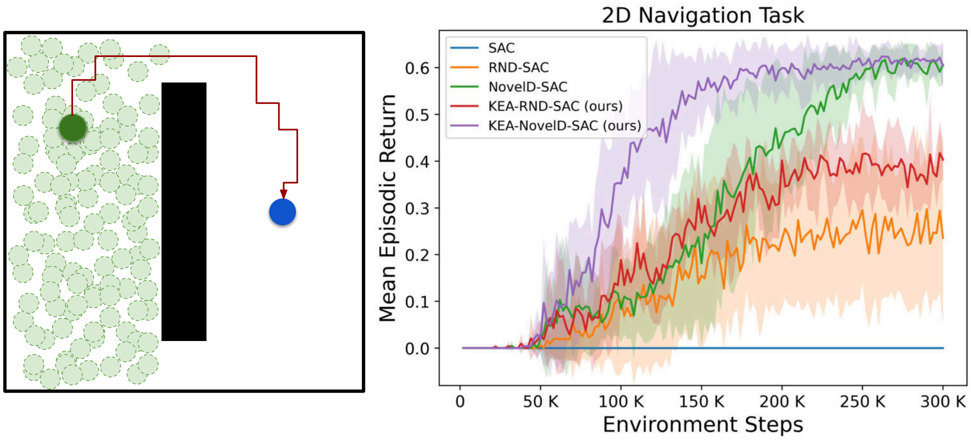


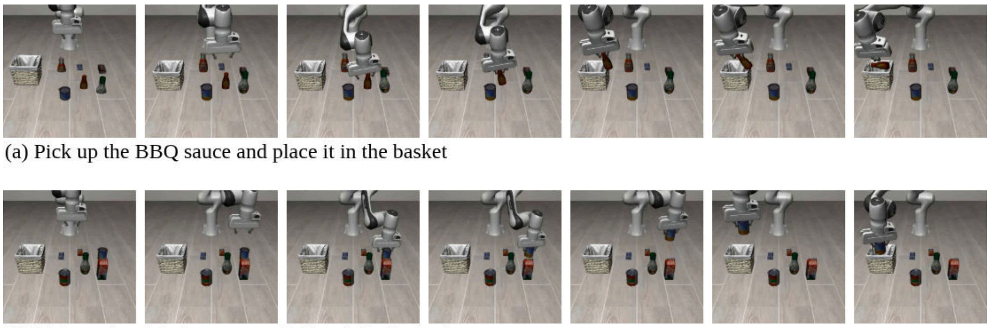
Figure 26: *Left*: 2D Navigation task involves navigating an agent from a randomly chosen start (light green circles) to a fixed goal position on the right (blue point) while avoiding an obstacle placed in the middle of the environment. *Right*: Mean episodic returns during training. The shaded area spans one standard deviation.

Algorithm	**DeepSea 30**
DeRL-A2C [46]	0.09 ± 0.08
DeRL-PPO [46]	-0.01 ± 0.01
DeRL-DQN [46]	0.10 ± 0.10
SOFE-A2C [47]	0.04 ± 0.09
SOFE-PPO [47]	0.09 ± 0.23
SOFE-DQN [47]	0.42 ± 0.33
SAC	0.00 ± 0.00
RND-SAC	0.35 ± 0.44
KEA-RND-SAC (ours)	**0.54 ± 0.32**

Table 3: Average performance on DeepSea environments of 30 × 30, reported with one standard deviation over 100,000 training episodes.

Method	**Walker Run**	**Cheetah Run**	**Reacher Hard**
SAC	0. ± 0.	0. ± 0.	715.17 ± 216.57
RND-SAC	287.65 ± 334.12	512.02 ± 466.26	790.32 ± 143.26
KEA-RND-SAC (ours)	**629.74** ± **196.75**	**773.76** ± **162.74**	**874.61** ± **94.58**
NovelD-SAC	553.26 ± 191.03	647.29 ± 382.58	**860.40** ± **76.15**
KEA-NovelD-SAC (ours)	**706.47** ± **389.23**	**734.67** ± **316.95**	837.12 ± 68.95

Table 4: Mean episodic return (mean±std.) in three sparse reward tasks from the DeepMind Control Suite.



(a) Pick up the BBQ sauce and place it in the basket

(b) Pick up the alphabet soup and place it in the basket

Figure 27: Examples of manipulation sequences generated by the proposed VLA model. Given different language instructions, the model successfully grounds visual input and executes task-specific actions. (a) The robot picks up the BBQ sauce and places it in the basket. (b) The robot picks up the alphabet soup and places it in the basket.

4.4 Pretraining VLA for Efficient Learning

Training robotic agents directly in the real world is costly and potentially unsafe, driving researchers to utilize simulation environments extensively. However, simulation training introduces a notable physics dynamics gap, particularly evident in complex manipulation tasks involving object-gripper interactions. This gap significantly hampers the transferability of learned skills from simulation to reality.

To address this challenge, we are currently exploring an innovative direction by leveraging a pre-trained vision-language model to learn robotic manipulation from large-scale robotic interaction data to form a **vision-language-action (VLA) model** [48]. This model is designed to predict actions based on visual observations and task descriptions (see task examples in fig. 27), enabling more effective and generalizable policy learning.

We introduce a hierarchical VLA architecture with distinct high-level and low-level components operating at different temporal resolutions. The high-level module interprets task descriptions and observations to guide overall execution, while the low-level module enables fast, reactive control for real-time performance. Furthermore, the foundation model's adaptability enables quick fine-tuning with minimal additional data, substantially reducing the resources required for real-world deployment.

Further evaluation is ongoing, including the effectiveness of our model compared to the state-of-the-art methods.

5 T2.3: In-Hand Grasp Perception

Task 2.3 focuses on reconstructing the state of the gripper-object system after performing a grasp, with the primary aim of providing crucial insights into the throwing controller developed in T4.4.

It also focuses on implementing sensors and sensing strategies to enhance the success rate of the grasping strategy with the SoftHand, which was developed in T4.3 for moving objects on a conveyor belt.

5.1 In-hand perception after grasping

The key contributions for task T2.3 concerning the reconstruction of the the state of the gripper-object system after performing a grasp, were already reported in D2.2 and include the following points, which we summarise below for completeness.

- **Tactile sensing to detect and prevent grasp failure:** A framework able to: 1) predict grasp failure with soft robotic hands exploiting tactile information, and 2) implement a set of reaction strategies to avoid object slippage from the hand [15].
- Simultaneous perception and manipulation for **estimating inertial parameters of in-hand objects**.
- Evaluating visual 6D pose estimation, involving novel objects with slight occlusion caused by a multi-finger hand.

One crucial point related to the in-hand perception is the evaluation of the grasp stability. While the usage of a soft underactuated gripper – like the SoftHand being used in the DARKO platform – permits handling the uncertainty related to the picking of an object in an easier way, the compliance of the end-effector fingers makes it difficult to predict the grasp configuration and consequently know in advance if the grasp is successful. Vision-based frameworks would fail in cases where a small object is heavily occluded by the gripper, and furthermore require the gripper to always be in the camera's field of view while manipulating the object. UNIPI [15] have developed a method to combine distributed tactile sensing and machine learning to detect sliding conditions for a soft robotic hand mounted on a robotic manipulator, targeting the prediction of the grasp failure event and the direction of sliding. The outcomes of these predictions allow for an online triggering of a compensatory action to prevent the failure. Despite the fact that the network was trained only with spherical and cylindrical objects, we demonstrate high generalization capabilities, achieving a correct prediction of the failure direction in 75 % of cases, and a 85 % of successful re-grasps, for a selection of twelve objects of common use.

Furthermore, it is important to estimate the inertial properties of the grasped object in order to plan a throwing action better, for cases where the object cannot be well approximated by a point mass. The goal here is to identify the optimal movements that a generic object should follow to maximize the accuracy and minimize the uncertainty for the estimation of its unknown mechanical properties: the mass, the three spatial center of mass terms and the six independent elements of the inertia tensor. We optimise the parameters of a continuous function with a continuous derivative (with a null derivative at each interval extremity) which minimises the maximum state uncertainty of these parameters. In experimental tests with a 3D-printed T-shaped object with known mass and density distribution, the centre of mass error norm was estimated to within 5 mm of ground truth, compared to approximately 20 mm when using random movements.

5.2 In-hand perception for pre-grasping of moving objects

The key contributions for task T2.3, which aimed to implement sensors and sensing strategies to improve the success rate of the grasping strategy using the SoftHand, were developed during the final year of the project and are briefly described below.

In [49], an active correction control sensor algorithm demonstrated promising results for static objects. However, each experiment required up to two minutes, making it unsuitable for scenarios with shorter time constraints, such as grasping moving objects. An alternative approach involves using sensors to achieve a reflex-like response mechanism that detects objects within the SoftHand's range and immediately executes the grasp. While this approach improved grasp performance for certain cases, it relied on a simplistic go/no-go decision. For this work, a new approach at an in-hand sensor implementation will be conducted to investigate the possibility of using these sensors to localize objects

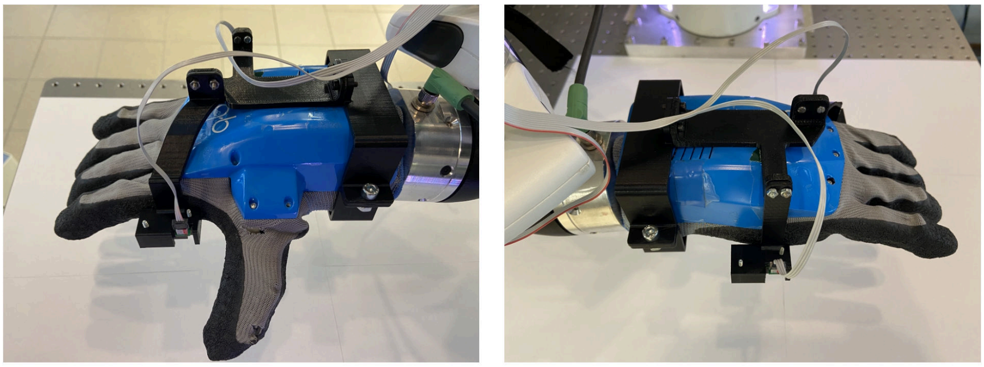


Figure 28: The specially designed sensor holder, attached onto the SoftHand to steady position the 2 proximity sensors in the desired configuration.

more precisely. This requires a sensor implementation that balances active control as in [49] with the simplicity of a binary contact/no-contact decision.

The proposed sensor implementation integrates the proximity sensors into the existing framework for grasping moving objects, as illustrated in D4.2, through an exploratory motion strategy. In this approach, the robot moves towards the object and executes a rotational exploratory motion with the SoftHand. This motion occurs at a fixed height relative to the table surface and rotates solely around the vertical z-axis, while the sensors actively collect data. The rotational exploratory motion can be clockwise or counterclockwise depending on the conveyor belt motion relative to the robot. By maintaining a consistent height, the sensor readings can distinguish between detecting an object and passing over the table surface.

The proposed implementation uses proximity sensors to detect the optimal grasping angle before the SoftHand executes the grasp on the object. For instance, consider the SoftHand rotating at a constant height of 0.15 m above the table surface. If the sensors detect a decrease in distance measurements to readings of 0.07 m, it indicates the presence of an object, such as a tennis ball with a diameter of 0.07 m. This exploratory motion provides essential information about the object's position relative to the SoftHand, enabling more informed grasping strategies.

A sensor holder, as shown in Figure 28, was developed. This holder, consisting of two clamping ring pieces, securely and robustly secures the sensors around the SoftHand's wrist.

Experiments were conducted to move objects using the sensors for 15 objects. Each object was tested 8 times, resulting in a total of 120 experiments. The results are presented in Table 5. These results show a total success rate of 71%, which is an improvement compared to the 60% success rate achieved without sensor implementation, as shown in Table 6.

When comparing the results of moving objects with and without the sensor implementation, some observations emerge.

- Spherical objects, such as tennis balls, potatoes, baseballs, apples, and tape rolls, as well as the Sponge, continue to have a high success rate, exceeding 75

- For non-spherical objects, a slight increase in success rate was observed, including the Pringles Can, courgette, block, and paper box.

- For challenging objects that previously had low success rates due to their orientation dependency and small diameter, the sensor implementation improved their success

#	Object	Class	Experiments	Success	Fail	Rate
1	Tennisbal	Power sphere	8	8	0	100%
2	Potato	Quadpod	8	7	1	88%
3	Spatula	Small diameter	8	4	4	50%
4	Deo	Medium	8	5	3	63%
5	Pringles	Large diameter or parallel extension	8	5	3	63%
6	Little ball	Tip pinch or Quadpod	8	3	5	38%
7	Sponge	Palmar pinch	8	8	0	100%
8	Block	Parallel extension	8	5	3	63%
9	Banana	Ring	8	3	5	38%
10	Courgette	Precision disk	8	6	2	75%
11	Baseball	Power sphere	8	7	1	88%
12	Can	Precision disk or parallel extension	8	5	3	63%
13	Box	Parallel extension	8	6	2	75%
14	Apple	Power sphere	8	7	1	88%
15	Tape roll	Power sphere or ring	8	6	2	75%
#	Total		120	85	35	71%

Table 5: Results for experiments on moving objects with in hand sensing, showing per object, the most common class that was detected for the object, the amount of experiments performed, the amount of successful and failed grasps, as well as the success rate.

rate. For example, the spatula's success rate increased from 0% to 50%, the deodorant can's success rate increased from 25% to 63%, the banana's success rate increased from 13% to 38%, and the little ball's success rate increased from 0% to 38%.

Figure 29 illustrates how the robot's SoftHand approaches the object, performs exploratory motions while following it on the conveyor belt, and successfully grasps it once the optimal orientation is detected. The top row shows minimal SoftHand rotation during the Courgette experiment, as the optimal orientation was detected quickly. The middle and bottom rows, however, showcase the Box and Pringles Can, demonstrating large rotations around the z-axis to determine the best grasping orientation. Additional snapshots from experiments are shown in Figure 30, highlighting the exploratory motions for non-spherical objects. The top and bottom rows show successful grasps of previously challenging objects, such as the deodorant can and spatula. While the overall grasping performance of the system improved from 60% to 71%, especially for challenging non-spherical objects with previously low success rates, some cases of failed grasps were still observed.

Key reasons for these failures were identified during the experiments as follows:

Inaccurate position or velocity estimates: Errors in position and velocity estimates from the point cloud and Kalman filter caused some failures. Objects with small diameter, such as the little ball and banana, proved difficult to track due to their size and the way the camera workspace was cropped to exclude the conveyor belt surface. Similarly, for larger objects such as the Pringles can, when parts of the object partly extended out of the selected camera workspace, the pose estimate would be off.

Trajectory deviations: Deviations between the robot's planned and executed trajectories caused the SoftHand to not exactly reach the desired position over the object. In some cases, these deviations were small enough to be compensated by the sensors, or the grasp was still successful due to the adaptable nature of the SoftHand. However, in a few instances, larger deviations led to failed grasps.

#	Object	Class	Experiments	Success	Fail	Rate
1	Tennis ball	Power sphere	8	8	0	100%
2	Potato	Quadpod	8	8	0	100%
3	Spatula	Small diameter	8	0	8	0%
4	Deodorant	Medium diameter	8	2	6	25%
5	Pringles	Large diameter or Parallel extension	8	4	4	50%
6	Little ball	Tip pinch	8	0	8	0%
7	Sponge	Palmar pinch or Parallel extension	8	8	0	100%
8	Block	Parallel extension	8	4	4	50%
9	Banana	Ring	8	1	7	13%
10	Courgette	Precision disk or Quadpod	8	5	3	63%
11	Baseball	Power sphere	8	8	0	100%
12	Can	Precision disk or Parallel extension	8	5	3	63%
13	Box	Parallel extension	8	4	4	50%
14	Apple	Power sphere	8	8	0	100%
15	Tape roll	Power sphere or	8	7	1	88%
#	Total		120	72	48	60%

Table 6: Results for experiments with moving objects without in hand sensing, showing per object the amount of performed experiments, successes, fails and the success rate.

Suboptimal orientation detection: In some cases, the sensors failed to detect the optimal orientation of the object during the exploratory motion. For objects such as the block, banana or can, the exploratory motion could be performed without meeting the sensor threshold. As a result, the SoftHand closed in attempt to still grasp the object without optimal orientation, which was often unsuccessful.

6 T2.4: Detecting Successful Throws

The module responsible for assessing the result of throwing objects was reported in D2.2. In summary, we follow a trajectory-centric approach, estimating the landing location of the thrown object relative to a target bin, based on observations from one of the robot's onboard Azure Kinect cameras.

As reported in D2.2, our tests on the data sets recorded in the project indicate that we reach 100% binary classification accuracy (hit or miss the target bin) for *short* throws (1.25 m from the reach of the arm, 2 m from the robot base) which is in line with the target for the main use case at BSH.

7 T2.5: Perception of Humans and their Poses

The fifth and final task of the perception work package WP2 deals with the real-time 3D detection and tracking of humans and their articulated 3D poses from the egocentric perspective of a mobile robot and its onboard sensors, thereby addressing DARKO's objective O2 on efficiency in human-robot co-production and enabling further downstream tasks including building maps of dynamics (T3.3), prediction of human motion and intents (T5.1), communication with human co-workers (T5.2) and reasoning about human-robot spatial interactions (T5.3), which serve as inputs to safe and risk-aware control (T4.2) and local and global motion planning (T6.1, T6.3).

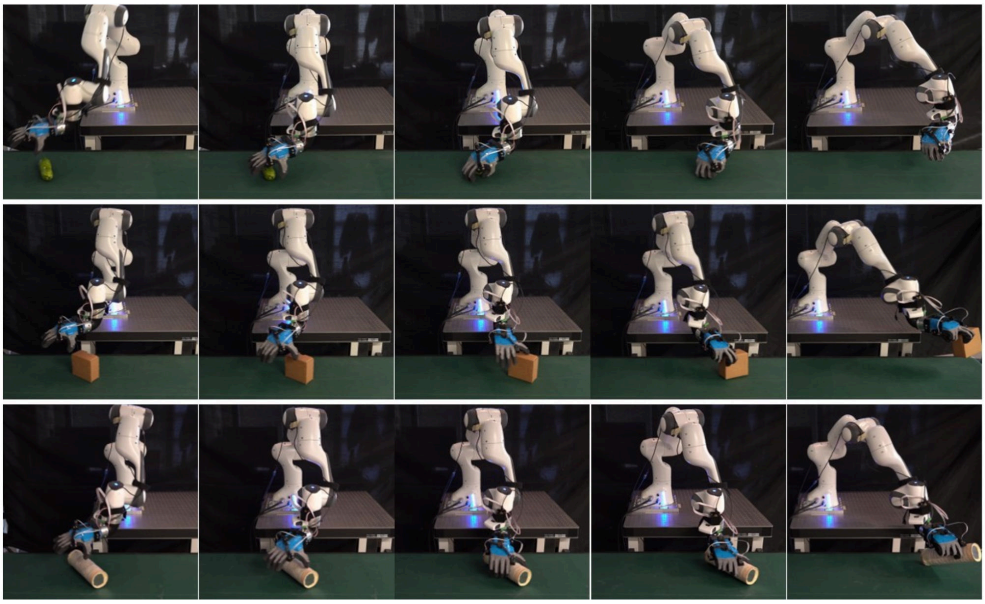


Figure 29: Snapshots taken during experiments for moving objects with the use of the sensors, clearly showing the exploratory motion of the SoftHand by rotating over the object until the optimal orientation to perform the grasp is detected. The used objects are the Courgette in the top row, the paper Box in the middle row and in the bottom row, the Pringles Can.

Key contributions in T2.5 (Reported in D2.2)

The following T2.5 contributions had already been reported as part of the initial perception system of DARKO in the preceding deliverable D2.2:

- A **systematic, cross-modal comparison** of human detection approaches for robotics, especially in intralogistics scenarios (IROS 2021).
- A complex multi-view recording setup leading to a **novel multi-modal dataset for 3D human pose estimation** with heterogeneous wide-FOV sensors, including fisheye cameras and 3D lidar.
- A fast ONNX & TensorRT implementation of the **RGB-D YOLO 3D human detector**, deployed on the DARKO robot as part of the MS2 milestone, and a baseline wide-FOV 3D human pose estimation approach that performs virtual pinhole reprojection of entire fisheye images.
- An **SVM-based method for data-efficient, single-frame, skeleton-based human activity classification** (“sitting”, “standing”, “kneeing”), demonstrated during milestone MS2.
- **UPTor**, a Transformer-based approach for joint short-term 3D human body pose and trajectory prediction (now accepted and recently presented at ICRA 2025).

Key contributions in T2.5 (since D2.2)

While task T2.5 was originally planned to finish with the submission of the preceding deliverable D2.2, the following projects and paper submissions that were already close to their completion have still been finalized in the final period to fully exploit the results from DARKO:

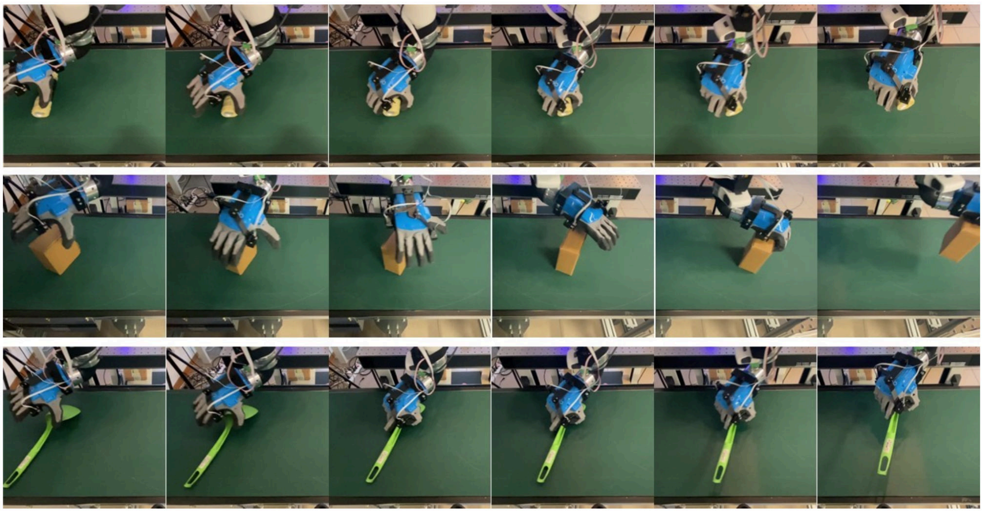


Figure 30: Snapshots taken during experiments for moving objects with the use of the sensors, clearly showing the exploratory motion of the SoftHand by rotating over the object until the optimal orientation to perform the grasp is detected. The used objects are the Deodorant can in the top row, the paper Box in the middle row and in the bottom row, the Spatula.

- Extension of the 3D human pose estimator MeTRAbs to natively support fisheye images **(Fisheye-MeTRAbs)**, using the novel multi-view evaluation dataset introduced in the previous deliverable D2.2, leading to a publication co-funded by DARKO at ICRA 2025.
- Integration of a state-of-the-art skeleton-based multi-frame activity recognition approach for **detecting dynamic gestures** that are relevant for intralogistics scenarios. A paper co-supervised via funding from DARKO, which compares the resulting method to VLM- and VFM-based approaches, has been accepted at RO-MAN 2025.
- Integration of the resulting methods with ROS and architectural extension towards **real-time 360-degree surround-view human perception** on the DARKO robot, with a showcase on robotic task specification as a downstream task (systems paper submission work in progress).
- Integration with T2.1 for object referral via **pointing gestures**.
- Further integrations of human perception with **context-aware MPC** from WP6 (RA-L 2024) and **Maps of Dynamics** from WP3, including live demonstrations during the final stakeholder meeting.

7.1 3D Human Pose Estimation on Fisheye Images

In the previous deliverable D2.2, Bosch had introduced a novel multi-view recording setup and multi-modal dataset with accurate 3D human body pose groundtruth, including also fisheye images with challenging close-up human poses, to foster research on wide field-of-view human perception in robotics applications.

To exploit these DARKO results from the previous period, as part of an ongoing collaboration with RWTH Aachen University supported by additional funding from Bosch outside of DARKO, the MeTRAbs [50] approach for absolute, metric-scale 3D human pose estimation has been extended with virtual pinhole crop reprojection as well as multiple fisheye-specific camera models. Virtual pinhole reprojection of detected *person crops* has computational advantages over the initially implemented DARKO baseline from D2.2 where we reproject the *entire fisheye image* to one or multiple virtual pinholes. However, both variants struggle when persons are close to the camera due to severe distortions. Instead, more generic models like the Double Sphere camera model by Usenko et al. [22] support fisheye projection with larger opening angles, as visualized in the qualitative example in Figure 31. We particularly found the **Double Sphere model** to be useful for robotics applications due to its fast, closed-form analytic inverse, and good generalization to all fisheye lenses that we experimented with in DARKO.

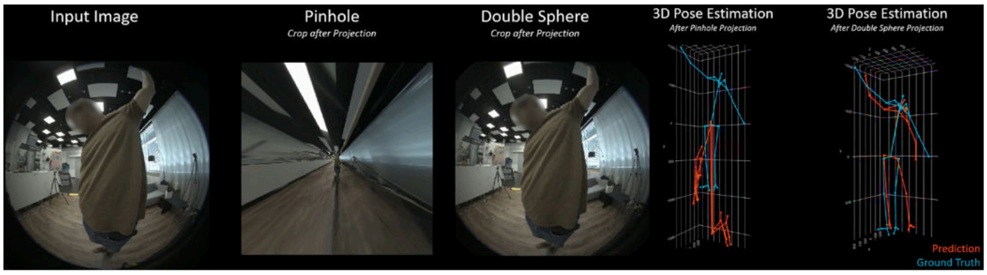


Figure 31: Fisheye lenses are particularly advantageous for perception of humans if the subject is close to the robot (0.5m distance to the camera, or less). The example here shows the raw fisheye input image, a virtual pinhole reprojection (which leads to extreme distortion, if the entire body pose shall be included in the resulting image), and a mild reprojection using a fixed-parameter Double Sphere [22] camera model. In the qualitative examples on the right, it can be seen that the projection using the Double Sphere model (rightmost picture) via our method proposed in [16] leads to a much more accurate 3D body pose compared to the multi-view groundtruth, than the result obtained using virtual pinhole reprojection.

This work on **Fisheye-MeTRAbs** has led to an ICRA 2025 publication [16], partly attributed to DARKO for the multi-view data recording setup from D2.2. In this work, different projection models for fisheye-based 3D human pose estimation have been systematically evaluated. The proposed architecture for 3D human pose estimation with fisheye or pinhole cameras is shown in Figure 32. One key finding is that the classic pinhole camera model works best when the human pose covers less than 120-130 degrees field of view. Instead, for larger opening angles (e.g. when the person is < 1 meter away from the camera), wide-FOV camera models such as the Double Sphere model by Usenko et al. lead to more accurate results. In the end, the work proposes a heuristic based on detected bounding box extents, for dynamically switching between these camera models, which has also been implemented on the DARKO robot.

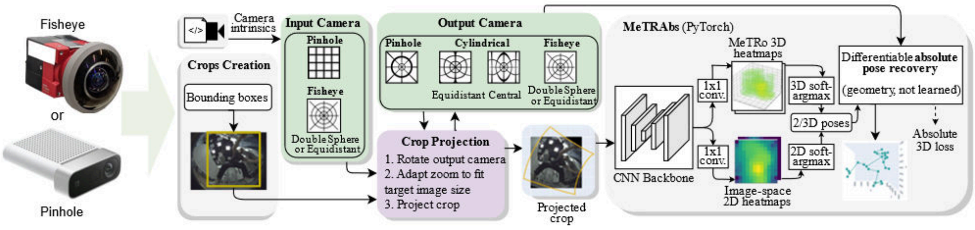


Figure 32: Architecture for absolute, metric-scale 3D human pose estimation in fisheye and pinhole images, as proposed in the ICRA 2025 publication [16], which extends [50].

For DARKO's final demonstration milestone, Bosch has developed the ROS integration of this new MeTRAbs implementation. This involved solving challenges related to the

ONNX export of the original Pytorch models, to allow for fast execution on the DARKO robot's GPU and Jetson hardware. One particular issue was that some of the algebraic operations required for absolute pose recovery in MeTRAbs, such as matrix inverse and least-squares optimization, are not supported by the current ONNX operation set. Here, we found that when excluding the absolute pose recovery step during export — which we now perform separately in Pytorch in a post-processing step, without any significant impact on runtime performance —, the backbone can successfully be exported to ONNX and then efficiently be executed using the TensorRT execution provider of Microsoft's ONNX Runtime, on either the Nuvo PC's or the Jetson's GPU at frame rates of 8-10 Hz.

7.2 360-Degree Surround View Human Perception Architecture

As part of the project's final demonstration, to conclude the research on wide-FOV perception of humans, Bosch has integrated the fisheye version of MeTRAbs into a 360-degree surround view human perception pipeline on the DARKO robot. The architecture of the resulting **surround-view 3D human pose estimation pipeline**, as deployed on the DARKO robot, is shown in Figure 33. A qualitative example showing 3 persons surround the robot can be seen in Figure 34. Higher-level modules such as dynamic gesture recognition or recognition of pointing gestures, described in the next subsections, subscribe to the output of this pipeline in the form of 3D skeletons, which are optionally associated over time via a nearest-neighbor centroid tracker [51] that is being run in parallel to this pipeline.

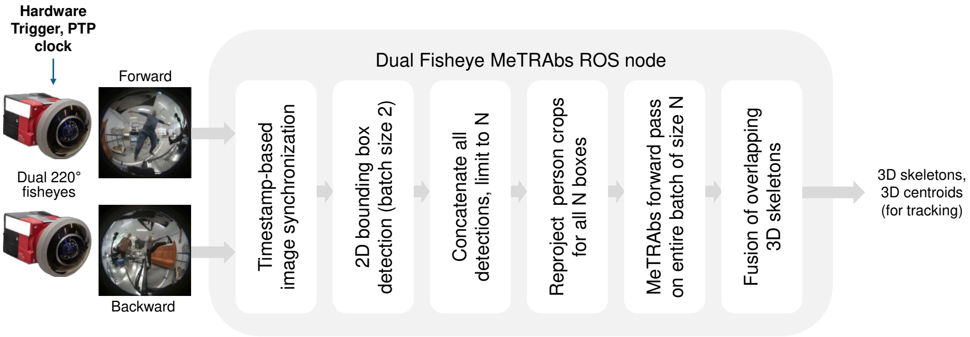

Hardware Trigger, PTP clock
Forward
Dual 220° fisheyes
Backward
Dual Fisheye MeTRAbs ROS node
Timestamp-based image synchronization
2D bounding box detection (batch size 2)
Concatenate all detections, limit to N
Reproject person crops for all N boxes
MeTRAbs forward pass on entire batch of size N
Fusion of overlapping 3D skeletons
3D skeletons, 3D centroids (for tracking)

Figure 33: Flow diagram and architecture of the dual fisheye 3D human pose estimation pipeline deployed on the DARKO robot for the final demonstration, based on [16]. A systems paper on this topic is currently in preparation for submission to RA-L.

During the MS4 stakeholder meeting, the fully integrated system performed well when moderate amounts of people were around the robot, reaching up to 8-10 Hz. In the presence of larger crowds, it was observed that the dynamic gesture recognition module (see next subsection) can suffer from low recall, due to frame rate dropping below 6-8 Hz, leading to too many frames of the characteristic gesture movements being dropped. As part of a planned final systems paper on human perception in DARKO, we want to further improve performance by speeding up person crop reprojection in the fisheye version of MeTRAbs, which likely is one of the bottlenecks here, via an optimized (e.g. C++-based) implementation. Details on the runtime performance of individual subcomponents will be reported in the planned final RA-L systems paper submission on surround-view human perception that is currently work in progress.

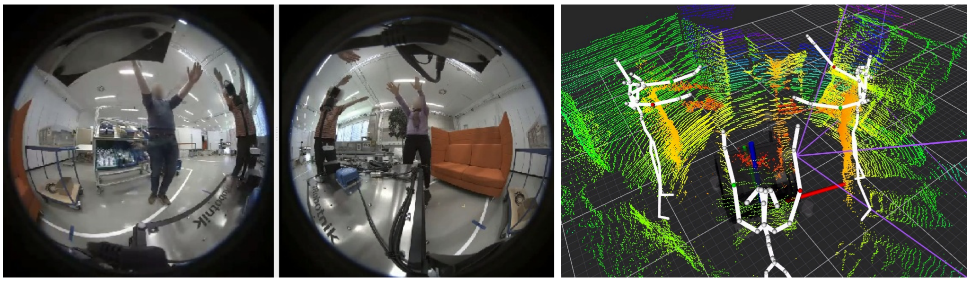


Figure 34: Our surround-view human perception system enables the robot to perceive humans also outside of the limited FOV of narrow-FOV cameras like the Azure Kinect, which – as visualized by the violet view frustum – would only perceive a single human in this example. Instead, our system can also estimate 3D human poses of the humans to the side and back of the robot. The 3D lidar point cloud is shown just for visualization purposes, our pipeline relies solely on fisheye images as input.

7.3 Recognition of Dynamic, Symbolic Gestures

As an exemplary downstream task that leverages results from 3D human pose estimation, in D2.2 we had introduced an SVM-based module for single-frame classification of human activities. This module did not yet consider any temporal motion cues. The recognized activities, as demonstrated during the MS2 milestone, included postures such as standing, sitting, kneeing or laying on the floor.

After MS2, Bosch had extended the training data with additional static, symbolic gestures (left/right arm stretched, T-pose, hands up) to symbolize different actions (driving left/right, continuing, stopping) that the user could trigger when interacting with the robot. Since the method used for obtaining skeletal input data (MeTRAbs [50]) does not provide fine-grained hand pose estimates, due to lack of finger joint annotations in the training data, the gestures rely on distinct full-body motions of e.g. the arms. While this may appear like a strong restriction, it can also be advantageous in mobile robotics applications where the user might stand several meters away from the robot, and thus image resolution might be quite limited (especially also with fisheye cameras), thus larger body parts can likely be recognized in a more robust fashion.

Experiments showed the following key issues with the system as implemented back then: 1.) False negatives due to limited FOV, e.g. when the user is standing very close or to the side of the robot, as visualized in Figure 35. This can also occur when the robot turns away from the user in response to a successfully recognized gesture. 2.) False positives due to the ambiguity of static gestures with non-gesture human activities, e.g. having a vivid discussion, answering a phone call, scratching one's head.

To resolve the first issue, we have – in the third period – integrated the system with the **fisheye-based surround-view 3D human pose estimation** architecture, as described in section 7.2. As demonstrated in experiments, this allows the DARKO robot to recognize gestures in varying distances and observation angles relative to the robot platform, including also cases where the robot is standing to the side of, or behind the robot.

To address the second point, we have switched from static to **dynamic gestures**, i.e. temporal sequences of characteristic movements, which can contain complex motion patterns and thus be less ambiguous than single-frame static gestures. In order to support temporal motion sequences as input, an existing state-of-the-art graph-based approach originally developed for action recognition (**HD-GCN** [52]) has been integrated by Bosch into the DARKO perception stack. As part of an external collaboration of Bosch with RWTH Aachen University, the method has been trained on a novel dataset comprised of 15

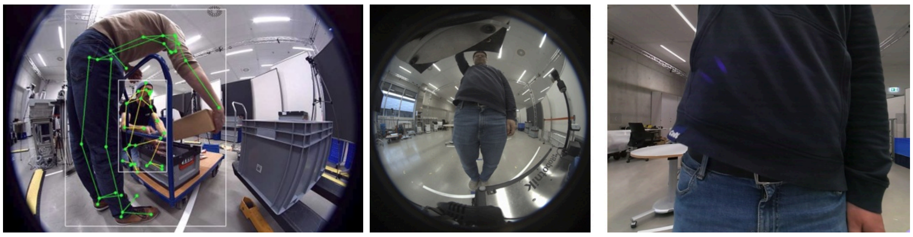
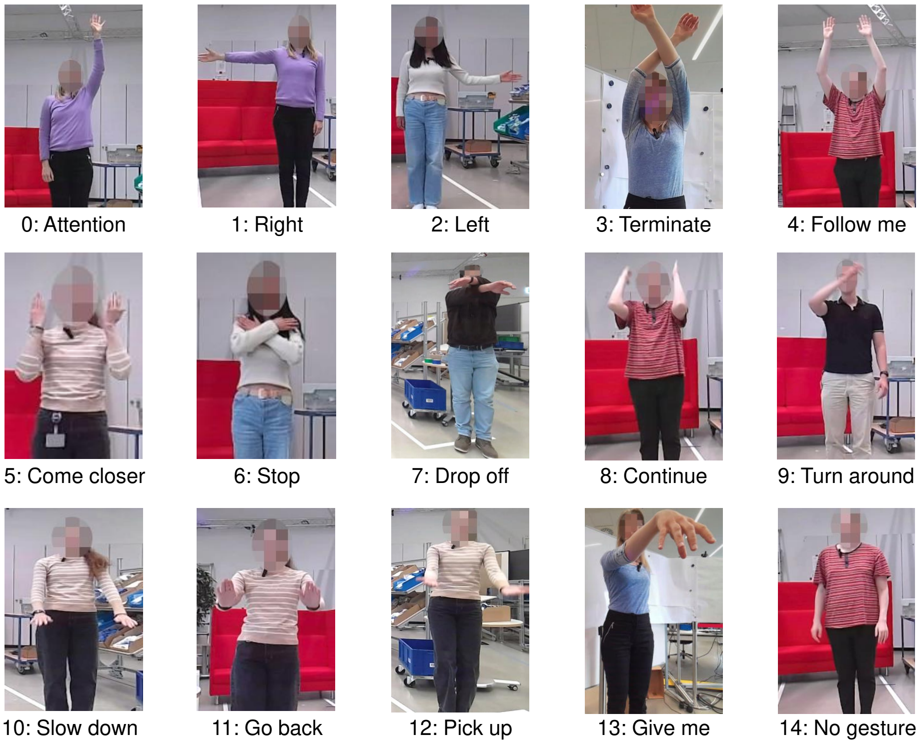


Figure 35: Surround-view human perception with wide-FOV fisheye lenses allows the robot to acquire rich scene context, as shown in the example on the left where two humans are loading boxes onto a cart, or the examples on the right where a user is performing the "Attention" gesture while standing close to the robot. In the fisheye case (middle picture), the gesture is easily recognizable, whereas in the Kinect pinhole image (right picture), there is a high amount of uncertainty on which gesture is being performed, if any.

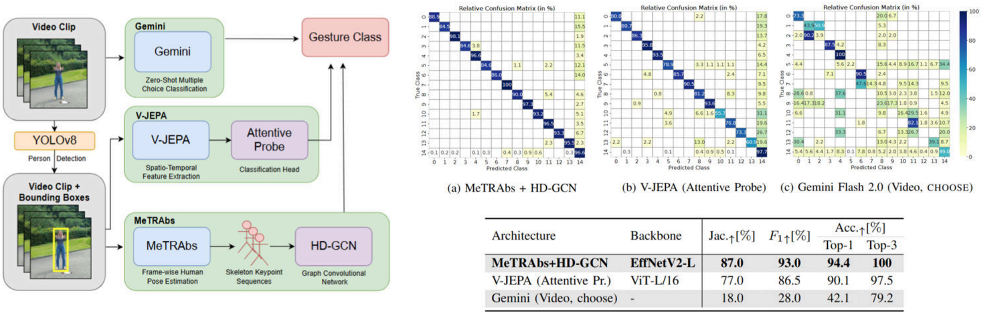
different full-body gestures aimed at mobile navigation and manipulation tasks, including gestures such go left, turn around, slow down, stop, continue, pick up / drop off / hand over item, pay attention, terminate interaction. Figure 36 provides an overview of the included gesture classes, including a background class for non-gestures.



0: Attention
1: Right
2: Left
3: Terminate
4: Follow me
5: Come closer
6: Stop
7: Drop off
8: Continue
9: Turn around
10: Slow down
11: Go back
12: Pick up
13: Give me
14: No gesture

Figure 36: List of 15 dynamic gestures designed for task specification for mobile manipulation tasks at medium to far distances. A skeleton-based method for recognition of these gestures based on fisheye 3D human pose estimates has been implemented on the DARKO robot.

This work has led to an accepted paper at RO-MAN 2025 [17], in which RWTH additionally compared the approach to a **fined-tuned vision foundation model (VFM)** and a **prompted vision-language model (VLM)**, with the skeletal approach performing



(a) MeTRAbs + HD-GCN (b) V-JEPA (Attentive Probe) (c) Gemini Flash 2.0 (Video, CHOOSE)

Architecture	Backbone	Jac.↑[%]	F_1↑[%]	Acc.↑[%]	
				Top-1	Top-3
MeTRAbs+HD-GCN	**EffNetV2-L**	**87.0**	**93.0**	**94.4**	**100**
V-JEPA (Attentive Pr.)	ViT-L/16	77.0	86.5	90.1	97.5
Gemini (Video, choose)	-	18.0	28.0	42.1	79.2

Figure 37: Left: Experimental setup in the paper accepted at RO-MAN 2025 [17], where a skeletal-based approach is compared to a fine-tuned vision foundation model (VFM) and a prompted vision-language model (VLM) on the task of dynamic gesture recognition with 15 gesture classes. Right: Experimental results obtained using the 3 different approaches, showing that the skeleton-based method performs best.

best, the VFM coming close, and the VLM struggling to distinguish the 15 distinct gesture classes (Figure 36) from just textual descriptions, as shown in Figure 37. The paper submission acknowledges support by DARKO for supervision of the PhD student by Bosch.

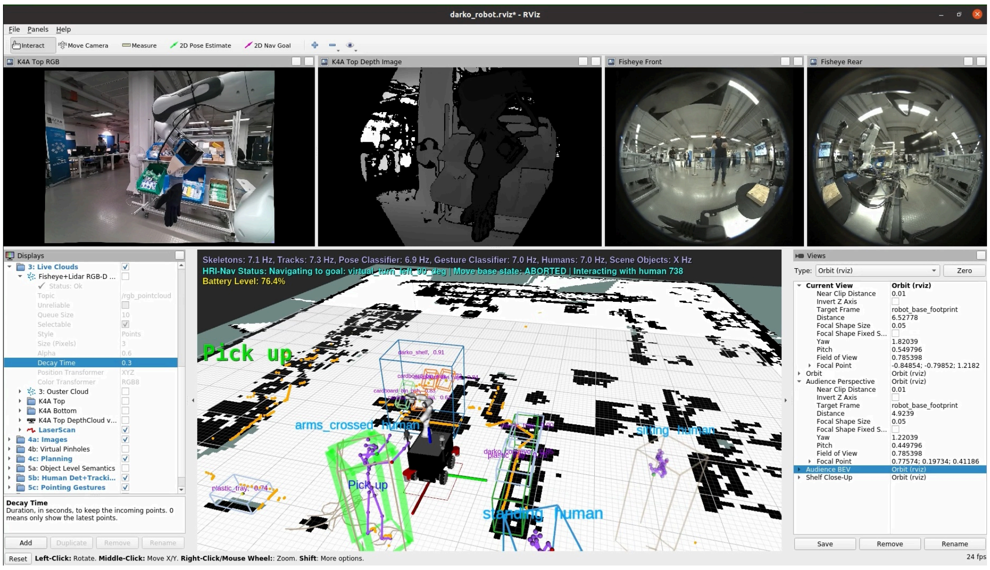
Bosch has deployed the integrated method on the DARKO robot and added support for using ONNX Runtime with the TensorRT execution provider (FP16 mode) for fast real-time performance. As a showcase for the project's **final demonstration** on the DARKO robot, Bosch has implemented a ROS-based demonstration where the robot provides verbal feedback on recognized gestures, and is able to perform exemplary actions such as navigating to a given direction, starting a manipulation sequence, or following a human. Figure 38 shows an Rviz view recorded during the final stakeholder meeting, where the user performs the "Pick up" gesture, after having gained the robot's attention using the "Attention" gesture (causing the robot to accept gestures only from that particular user, highlighted by the green bounding box). In response to the "Pick up" gesture, the robot drives to a point where it can observe the shelf (using a method from T2.1), detect the source bin for picking and estimate its 9DoF pose, before then navigating to a more close-up position from where the robot can then pick up an item from the shelf.

7.4 Pointing Gestures & Integration with Object-Level Semantics

To showcase that we can also understand **interaction between objects (T2.1) and humans (T2.5)** in real-time, we developed a system which combines human and object perception to recognize pointing gestures. This capability could be used for intuitive task specification, for instance, where a user points to an item to be picked up or to a tray for the DARKO robot to place or throw an item.

The method leverages real-time 3D skeleton tracking from the human perception module (T2.5) and 9DoF bounding boxes estimated by the object-level semantics module (T2.1). A primary pointing vector is estimated from the user's skeleton, originating from the elbow joint and passing through the wrist. To account for inherent inaccuracies in joint estimation and the ambiguity of a pointing gesture, we model the pointing direction not as a single ray, but as a narrow cone. Multiple rays are then stochastically sampled within this cone and traced into the scene. An object is identified as the potential target if it is consistently intersected by these rays over a short period of time. The closest of these consistently intersected objects is then selected as the final target of the gesture.

Figure 39 illustrates successful examples of the system in action. Generally, our experiments showed this approach to be effective for larger well separated objects, but



Skeletons: 7.1 Hz, Tracks: 7.3 Hz, Pose Classifier: 6.9 Hz, Gesture Classifier: 7.0 Hz, Humans: 7.0 Hz, Scene Objects: X Hz
HRI-Nav Status: Navigating to goal: virtual_turn_left_90_deg | Move base state: ABORTED | Interacting with human 738
Battery Level: 76.4%

3: Live Clouds	☑
Fisheye+Lidar RGB-D ...	□
✓ Status: Ok	
Topic	/rgb_pointcloud
Unreliable	□
Queue Size	10
Selectable	☑
Style	Points
Size (Pixels)	3
Alpha	0.6
Decay Time	0.3
Position Transformer	XYZ
Color Transformer	RGB8
3: Ouster Cloud	□
K4A Top	□
K4A Bottom	□
K4A Top DepthCloud v...	□
LaserScan	☑
4a: Images	☑
4b: Virtual Pinholes	□
4c: Planning	☑
5a: Object Level Semantics	□
5b: Human Det+Tracki...	☑
5c: Pointing Gestures	☑

Decay Time
Duration, in seconds, to keep the incoming points. 0 means only show the latest points.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

Type: Orbit (rviz)

Current View	**Orbit (rviz)**
Near Clip Distance	0.01
Invert Z Axis	□
Target Frame	robot_base_footprint
Distance	6.52778
Focal Shape Size	0.05
Focal Shape Fixed S...	□
Yaw	1.82039
Pitch	0.549796
Field of View	0.785398
Focal Point	-0.84854; -0.79852; 1.2182
Orbit	Orbit (rviz)
Audience Perspective	Orbit (rviz)
Near Clip Distance	0.01
Invert Z Axis	□
Target Frame	robot_base_footprint
Distance	4.9239
Focal Shape Size	0.05
Focal Shape Fixed S...	□
Yaw	1.22039
Pitch	0.449796
Field of View	0.785398
Focal Point	0.77574; 0.19734; 0.41186
Audience BEV	Orbit (rviz)
Shelf Close-Up	Orbit (rviz)

Figure 38: Screenshot from the final stake holder meeting demonstration, where the user observed by the frontal fisheye is performing a "pick-up" gesture (with the arms symbolizing a forklift that is lifting up an item) that is being recognized by the robot (green text). The green bounding box around the human indicates that this human has performed an "Attention" gesture earlier on, which, as a simple method of engagement detection, activates the recognition of further gestures by this user. The entire system, including fisheye-based 3D human pose estimation, was running in real-time on the DARKO robot.

we also identified some limitations: the accuracy of the gesture recognition is highly sensitive to the precision of the 3D joint estimation. Even small errors in the estimated 3D positions of the elbow and wrist joints can result in significant deviations of the pointing cone, especially over distance. This makes it challenging to reliably select smaller or more cluttered objects, such as individual bins on a shelf, without additional visual feedback for the user to confirm the selection. Nevertheless, this work demonstrates a successful integration where the human and object perception modules interact in real-time on the robot's hardware to enable a more natural form of human-robot collaboration.

7.5 Integration with Downstream Prediction and Planning Components

The human perception pipeline has been integrated with several downstream modules from DARKO's work packages WP3 (mapping), WP5 (human-robot spatial interaction) and WP6 (planning for the mobile base).

During the final stakeholder meeting demonstration, we have shown how the outputs from human perception can be used to perform live updates of **Maps of Dynamics**: In Figure 40, the human skeleton on the left is walking around in the scene, while the audience is watching. The map of dynamics, visualized by the arrows that indicate human flow direction, gets updated in an online fashion, visualized in the associated video animation by varying arrow directions, and new arrows being added.

In the initial report on perception (D2.2), we had introduced an efficient Transformer-based approach for short-term 3D human body pose and trajectory prediction, which can be used to improve temporal accuracy of skeleton estimates over time especially during temporary occlusions. This collaboration with DARKO WP5 has led to an accepted paper presented at ICRA 2025 [18]; Figure 41 provides an overview of the **UPTor** approach.

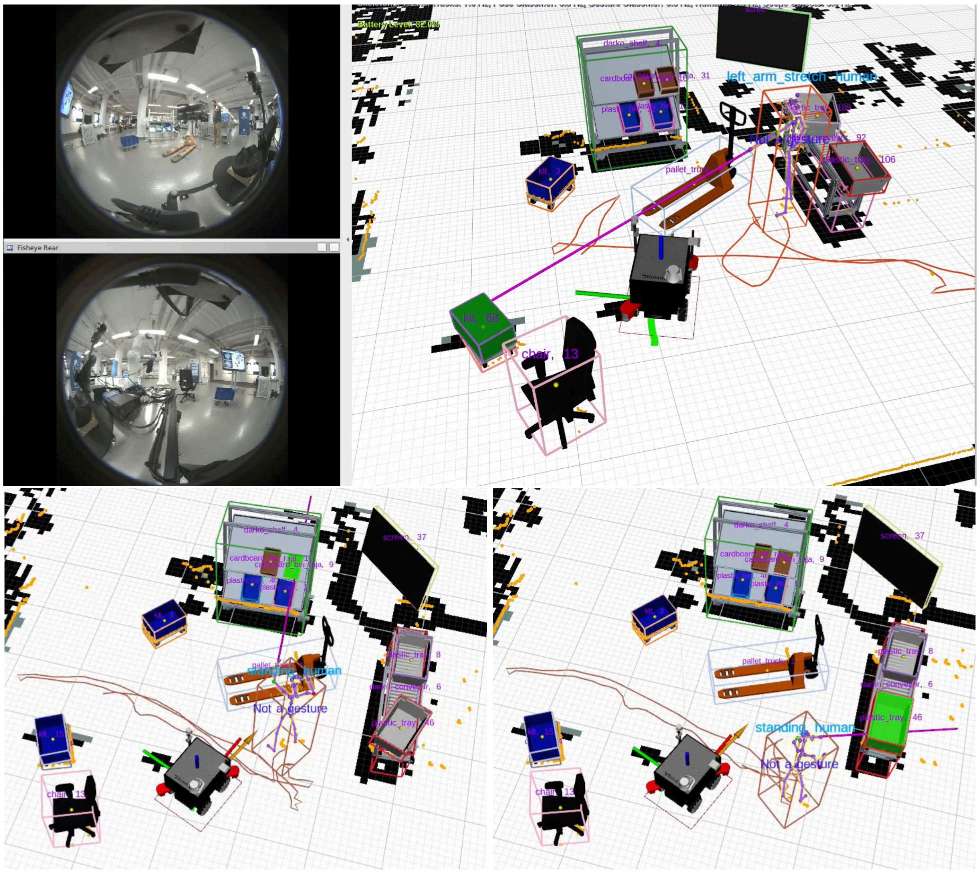


Figure 39: RViz screenshots demonstrating pointing gesture recognition system running live on the DARKO platform during final integration at KI.Fabrik in Munich. The main panels show 3D representation of the scene with detected objects and a tracked human skeleton (three different examples are provided). A pointing vector (magenta ray) is estimated from the user's arm to select a target object, which is highlighted in green. The side panels display the corresponding live video streams from the robot's front and rear fisheye cameras.

Finally, the 3D human pose estimation stack has been successfully integrated with a **context-aware MPC** approach for local motion planning of the mobile platform. This joint work with DARKO WP6 has led to a RA-L article [19], which leveraged the pinhole variant of the 3D human perception stack and our SVM-based classifier to detect e. g. people lying on the ground, and adapt the robot's behavior accordingly, as shown in Figure 42. During the final stakeholder meeting, we further demonstrated the integration with the fisheye-based 3D human pose estimation stack.

8 Conclusion

In this deliverable, we have presented the final perception system developed for DARKO, including advanced novel methods, datasets, experimental results and demonstration showcases for broad-level scene understanding (tasks T2.1 and T2.5) and perception for manipulation and throwing (T2.2, T2.3, T2.4).

Key highlights for broader-level scene understanding of objects (using LiDAR + fisheyes) and humans (using only fisheye cameras) include a surround-view real-time perception

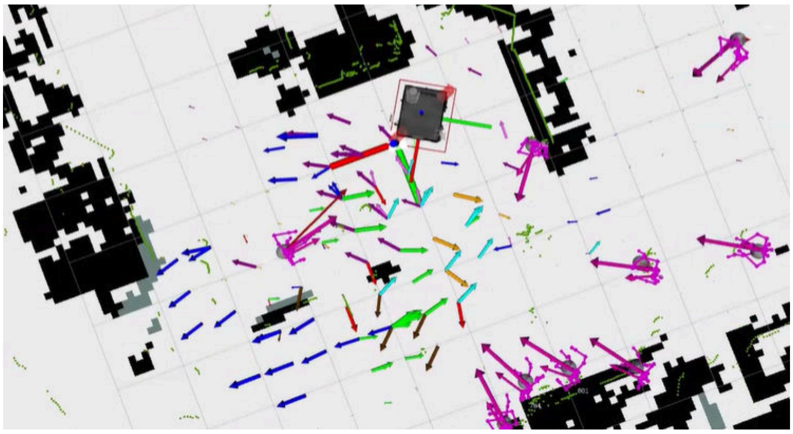


Figure 40: Using outputs from the T2.5 surround-view 3D human perception pipeline, we have performed a live demonstration of online update functionality for Maps of Dynamics (WP3) during the final stakeholder meeting at KI.Fabrik, Munich.

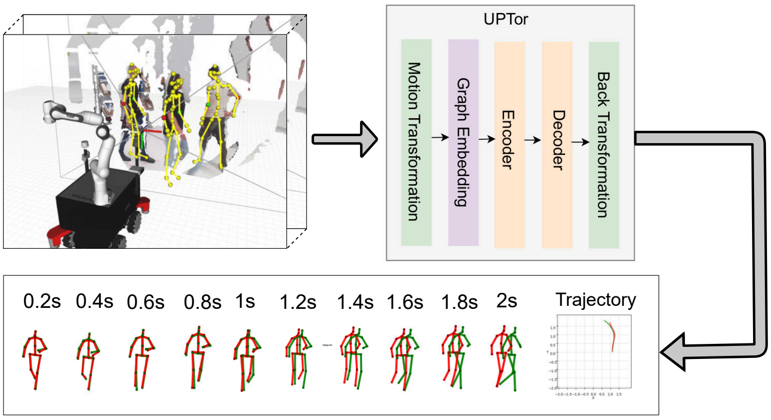
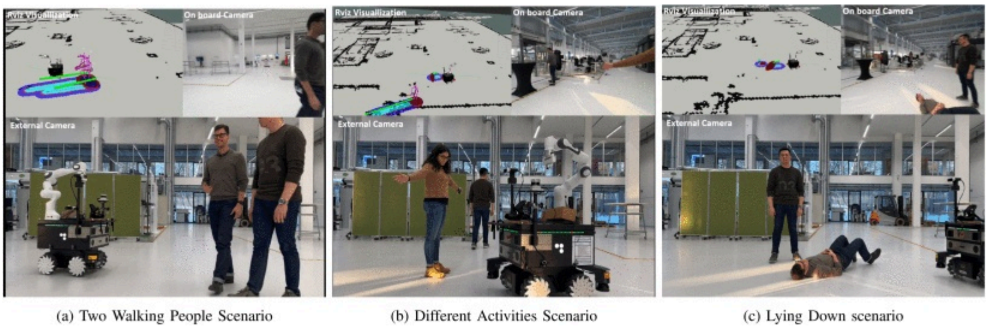


Figure 41: Integration of 3D human perception with a module developed together with WP5 for joint prediction of 3D body poses and trajectories (UPTor, accepted at ICRA 2025 [18]).



(a) Two Walking People Scenario (b) Different Activities Scenario (c) Lying Down scenario

Figure 42: Integration of 3D human perception and body pose classification with context-aware model predictive control from WP6 for human-aware navigation, accepted at RA-L 2024 [19].

pipeline, as well as methods for higher-level reasoning, e. g. using open-vocabulary 3D scene graphs and semantic neural radiance fields for relational modelling, or approaches for the detection of dynamic, symbolic gestures in mobile manipulation scenarios.

Key highlights for perception for manipulation include novel methods for regressing feasible grasps of unknown objects from sensor data, in a way that is transferrable from two-fingered rigid grippers to five-fingered underactuated soft hands, hierarchical reinforcement learning of parameterised motion primitives for configurations that do not afford direct grasp acquisition but instead require sequences of actions to be picked, a generally applicable method for efficient reinforcement learning in sparse reward settings (common in manipulation but also many other applications of reinforcement learning), and methods for in-hand perception to detect and correct slippage as well as estimating the inertial properties of grasped objects.

Intensive focus has been put by the consortium on experiments with actual data or objects from the industrial target domain of DARKO's lead use-case (as opposed to standard datasets from e. g. household scenarios which are more commonly publicly available).

A further focus was on the efficient real-time deployment of an integrated system on a resource-constrained mobile robot, This work includes calibration, e. g. by evaluating which camera models work best on the fisheye cameras of the DARKO platform, and introducing new approaches to acquire blur-free calibration images, as well as measures to maximize performance, e. g. by identifying bottlenecks in terms of consumed network bandwidth, adding efficient image transport mechanisms, and increasing CPU performance by installing additional fans. This integration and deployment work is technically very challenging, but at the same time brings the results closer to practical application in future robotic products.

Altogether, these results have paved the way towards a successful final project demonstration of the integrated DARKO robot platform at the final milestone demonstration (MS4 at KI Fabrik in Deutsches Museum, Munich) with a live audience of invited stakeholders.

9 References

WP2 publications

[1] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. “Open3DSG: Open-Vocabulary 3D Scene Graphs from Point Clouds with Queryable Objects and Open-Set Relationships”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024.

[2] Sebastian Koch, Johanna Wald, Mirco Colosi, Narunas Vaskevicius, Pedro Hermosilla, Federico Tombari, and Timo Ropinski. “RelationField: Relate Anything in Radiance Fields”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025.

[3] Saumya Saxena, Blake Buchanan, Chris Paxton, Bingqing Chen, Narunas Vaskevicius, Luigi Palmieri, Jonathan Francis, and Oliver Kroemer. *GraphEQA: Using 3D Semantic Scene Graphs for Real-time Embodied Question Answering*. 2024. arXiv: 2412.14480 [cs.RO].

[4] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. “Context-aware grasp generation in cluttered scenes”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 1492–1498.

[5] Shih-Min Yang, Martin Magnusson, Johannes A Stork, and Todor Stoyanov. “KEA: Keeping Exploration Alive by Proactively Coordinating Exploration Strategies”. In: *Int. Conf. on Machine Learning (ICML)*. 2025.

[6] Timm Linder, Kadir Yilmaz, David B. Adrian, and Bastian Leibe. “Acquisition of high-quality images for camera calibration in robotics applications via speech prompts”. In: *German Robotics Conference*. 2025. arXiv: 2504.11031 [cs.RO].

[7] Rishabh Jain, Narunas Vaskevicius, and Thomas Brox. “Towards Self-Supervised Pre-Training of 3DETR for Label-Efficient 3D Object Detection”. In: *Workshop on Transformers for Vision at IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[8] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. “Auto3DSG: Autoencoding for 3D Scene Graph Learning via Object-Level Scene Reconstructiion”. In: *ICCV 2023 Workshop on Scene Graphs and Graph Representation Learning (SG2RL)*. 2023.

[9] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. “SGRec3D: Self-Supervised 3D Scene Graph Learning via Object-Level Scene Reconstruction”. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2024.

[10] Sebastian Koch, Pedro Hermosilla, Narunas Vaskevicius, Mirco Colosi, and Timo Ropinski. “Lang3DSG: Language-based contrastive pre-training for 3D Scene Graph prediction”. In: *International Conference on 3D Vision (3DV)*. 2024.

[11] Yash Goel, Narunas Vaskevicius, Luigi Palmieri, Nived Chebrolu, and Cyrill Stachniss. “Predicting Dense and Context-aware Cost Maps for Semantic Robot Navigation”. In: *IROS 2022 Workshop on Perception and Navigation for Autonomous Robotics in Unstructured and Dynamic Environments (PNARUDE)*. 2022.

[12] Yash Goel, Narunas Vaskevicius, Luigi Palmieri, Nived Chebrolu, Kai Oliver Arras, and Cyrill Stachniss. “Semantically Informed MPC for Context-Aware Robot Exploration”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023.

[13] Shih-Min Yang, Martin Magnusson, Johannes A. Stork, and Todor Stoyanov. "Learning Extrinsic Dexterity with Parameterized Manipulation Primitives". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2024.

[14] Shih-Min Yang, Martin Magnusson, Johannes Andreas Stork, and Todor Stoyanov. "Data-driven Grasping and Pre-grasp Manipulation Using Hierarchical Reinforcement Learning with Parameterized Action Primitives". In: *IROS 2023 Workshop on Leveraging Models for Contact-Rich Manipulation*. 2023.

[15] Giuseppe Averta, Federica Barontini, Irene Valdambrini, Paolo Cheli, Davide Bacciu, and Matteo Bianchi. "Learning to Prevent Grasp Failure with Soft Hands: From Online Prediction to Dual-Arm Grasp Recovery". In: *Advanced Intelligent Systems* 4.3 (2022).

[16] Stephanie Käs, Sven Peter, Henrik Thillmann, Anton Burenko, David Benjamin Adrian, Dennis Mack, Timm Linder, and Bastian Leibe. "Systematic Comparison of Projection Methods for Monocular 3D Human Pose Estimation on Fisheye Images". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2025.

[17] Stephanie Käs, Anton Burenko, Louis Markert, Õnur Alp Çulha, Dennis Mack, Timm Linder, and Bastian Leibe. "How do Foundation Models Compare to Skeleton-Based Approaches for Gesture Recognition in Human-Robot Interaction?" In: *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2025.

[18] Nisarga Nilavadi, Andrey Rudenko, and Timm Linder. "UPTor: Unified 3D Human Pose Dynamics and Trajectory Prediction for Human-Robot Interaction". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2025.

[19] Elisa Stefanini, Luigi Palmieri, Andrey Rudenko, Till Hielscher, Timm Linder, and Lucia Pallottino. "Efficient Context-Aware Model Predictive Control for Human-Aware Navigation". In: *IEEE Robotics and Automation Letters* 9.11 (2024).

Other references

[20] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. *TR3D: Towards Real-Time Indoor 3D Object Detection*. 2023. arXiv: 2302.02858 [cs.CV].

[21] Timo Röhling and Fraunhofer FKIE. *RTSP Image Transport for ROS*. https://github.com/fkie/rtsp_image_transport. 2021.

[22] Vladyslav Usenko, Nikolaus Demmel, and Daniel Cremers. "The Double Sphere Camera Model". In: *International Conference on 3D Vision (3DV)*. 2018. eprint: http://arxiv.org/abs/1807.08957.

[23] J. Wilm and E.R. Eiriksson. *calib.io Camera Calibrator software*. 2018.

[24] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. "General, Single-shot, Target-less, and Automatic LiDAR-Camera Extrinsic Calibration Toolbox". In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. 2023.

[25] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *INTERSPEECH* (2023).

[26] Timm Linder, Kilian Y. Pfeiffer, Narunas Vaskevicius, Robert Schirmer, and Kai O. Arras. "Accurate Detection and 3D Localization of Humans Using a Novel YOLO-Based RGB-D Fusion Approach and Synthetic Training Data". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.

[27] Jens Piekenbrinck and Christian Schmidt and Alexander Hermans and Narunas Vaskevicius and Timm Linder and and Bastian Leibe. "OpenSplat3D: Open-Vocabulary 3D Instance Segmentation using Gaussian Splatting". In: *Proceedings of the CVPR 2025 Workshop on OpenSUN3D*. June 2025.

[28] Tianhe Ren et al. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. 2024. arXiv: 2401.14159 [cs.CV].

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. "Segment Anything". In: *arXiv:2304.02643* (2023).

[30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: *arXiv preprint arXiv:2303.05499* (2023).

[31] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. "Deep Hough Voting for 3D Object Detection in Point Clouds". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[32] Charles R. Qi, Xinlei Chen, Or Litany, and Leonidas J. Guibas. "ImVoteNet: Boosting 3D Object Detection in Point Clouds With Image Votes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[33] Hao Yang, Chen Shi, Yihong Chen, and Liwei Wang. "Boosting 3D Object Detection via Object-Focused Image Fusion". In: *arXiv preprint arXiv:2207.10589* (2022).

[34] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. "Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 13154–13164.

[35] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[36] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. *3D Gaussian Splatting for Real-Time Radiance Field Rendering*. 2023. arXiv: 2308.04079 [cs.GR].

[37] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. "Scaling open-vocabulary image segmentation with image-level labels". In: *European Conference on Computer Vision*. Springer. 2022, pp. 540–557.

[38] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. 2023. arXiv: 2305.06500 [cs.CV].

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[40] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. "Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V". In: *arXiv preprint arXiv:2310.11441* (2023).

[41] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. In: *Int. Conf. on Machine Learning (ICML)*. PMLR. 2018, pp. 1861–1870.

[42] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. “Exploration by random network distillation”. In: *arXiv preprint arXiv:1810.12894* (2018).

[43] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. “Noveld: A simple yet effective exploration criterion”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25217–25230.

[44] Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvári, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado van Hasselt. “Behaviour Suite for Reinforcement Learning”. In: *International Conference on Learning Representations*. 2020.

[45] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. “Deepmind control suite”. In: *arXiv preprint arXiv:1801.00690* (2018).

[46] Lukas Schäfer, Filippos Christianos, Josiah P. Hanna, and Stefano V. Albrecht. “Decoupled Reinforcement Learning to Stabilise Intrinsically-Motivated Exploration”. In: *Adaptive Agents and Multi-Agent Systems*. 2021.

[47] Roger Creus Castanyer, Joshua Romoff, and Glen Berseth. “Improving intrinsic exploration by creating stationary objectives”. In: *International Conference on Learning Representations* (2024).

[48] Moo Jin Kim et al. “OpenVLA: An Open-Source Vision-Language-Action Model”. In: *arXiv preprint arXiv:2406.09246* (2024).

[49] Emanuele Luberto, Yier Wu, Gaspare Santaera, Marco Gabiccini, and Antonio Bicchi. “Enhancing Adaptive Grasping Through a Simple Sensor-Based Reflex Mechanism”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1664–1671.

[50] István Sárándi, Timm Linder, Kai O. Arras, and Bastian Leibe. “MeTRAbs: Metric-Scale Truncation-Robust Heatmaps for Absolute 3D Human Pose Estimation”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 3.1 (2021), pp. 16–30.

[51] T. Linder, S. Breuers, B. Leibe, and K. O. Arras. “On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016.

[52] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. “Hierarchically Decomposed Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017274