From Perception to Assistance: Open-Vocabulary Shared Autonomy for Robotic Manipulation

***Abstract*—Teleoperating a robotic manipulator in industrial environments demands precision that camera-based interfaces alone struggle to deliver. The operator must align the end-effector with a target in clutter, under limited depth perception, and without colliding with the surrounding structures. This paper presents a shared-autonomy framework that assists the operator throughout this process. A single RGB-D camera captures the operator's arm motion and hand gestures without wearables, fiducials, or a calibration stage. The intended target is specified by a free-form text prompt, grounded by a vision-language model in the robot's gripper camera, and tracked across its onboard cameras by a promptable video-segmentation model, resulting in a grasp frame continuously separated from the obstacle map. Every commanded motion is executed by a GPU-accelerated model-predictive controller that enforces self- and environment-collision avoidance against an online volumetric reconstruction, while a potential field corrects the operator's reference toward the grounded target during the final approach. An autonomous mode can be gesture-triggered to complete the grasp on the same target without a separate perception pipeline. The framework is validated on a quadruped mobile manipulator. The interface achieves a positional RMSE of 59 mm relative to motion-capture ground truth, and the controller keeps the arm at least 18 cm from obstacles while the operator deliberately commands the arm into them by 6 cm. In an industrial valve manipulation and a pick-and-place task, the full framework succeeded in all trials, while ablating either the collision or the assistance module produced failures through complementary mechanisms, and autonomous execution succeeded in four of five trials per task.**

***Index Terms*—Manipulators, Telerobotics, User interfaces, Intelligent systems, Human in the loop**

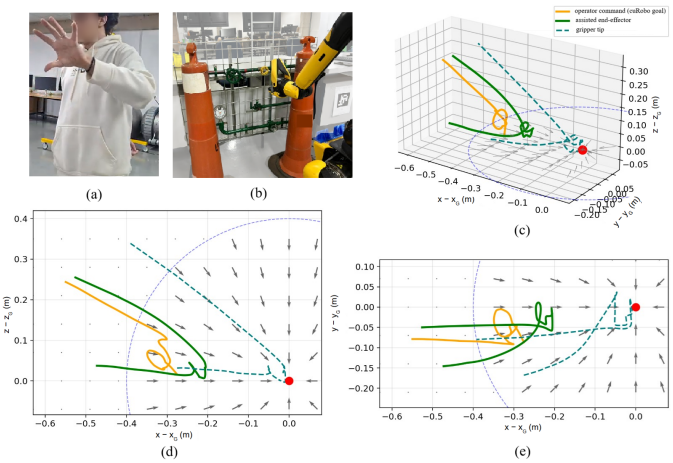

Fig. 1: Shared-control valve manipulation. (a) The operator commands the robot via arm motion and hand gestures, captured by an RGB-D camera without wearables or calibration. (b) The robotic arm operates on an industrial panel in clutter. (c)–(e) Approach trajectories relative to the grasp frame $\{G\}$ (red), grounded from the prompt "wheel valve". Inside the 0.4 m activation radius (dashed circle), the potential field draws the assisted end-effector trajectory (green) toward the target while the operator command (orange) remains offset. The gripper tip (dashed) converges onto the target, and the loops correspond to the operator turning the valve after the grasp.

I. Introduction

Quadruped mobile manipulators are increasingly employed for industrial inspection and intervention [1]–[3], which requires robots to navigate cluttered environments and interact with assets such as valves and tools. While the ability to traverse unstructured terrain and simultaneously execute dexterous tasks makes legged mobile manipulation uniquely suited to industrial settings, reliable manipulation in realistic workspaces remains challenging, as the system must couple locomotion stability, online perception in clutter, and contact-rich interaction within a unified framework [4]–[7].

For manipulation in such environments, both pure teleoperation and full autonomy have important limitations. Teleoperation is flexible, but it places a high cognitive burden on the operator, who must simultaneously command a high-DoF arm, manage viewpoint and clearance, avoid collisions, and execute contact-rich motions under limited feedback [8]–[11]. Fully autonomous systems can execute scripted behaviors, but they often depend on reliable perception, precise extrinsics, known objects, and task parameters that are difficult to guarantee under domain shifts in unstructured environments [12], [13]. Shared-control addresses this gap by allowing the operator to specify intent while assistive modules handle local motion generation and safety through corrective interventions or arbitration, improving success rates and reducing collisions and user effort [14]–[18].

However, many shared-autonomy systems still rely on assumption-heavy target-selection pipelines (known objects, markers, predefined affordances) and low-rate collision representations, which limit safety and responsiveness in real industrial scenes [14]–[16], [19]. For shared-control to remain usable during manipulation, the robot must perceive the environment onboard, continuously update a local collision representation, and react at interactive rates during execution, particularly for legged platforms whose advantage is operating outside fixed workcells. Recent onboard Truncated Signed Distance Field (TSDF)/Euclidean Signed Distance Function (ESDF) pipelines and learned distance-field representations make this level of continuous collision awareness increasingly practical [20], [21].

A further challenge is semantic grounding and task parameterization. Predefined object classes, fiducials, and affordance templates do not generalize well across industrial sites. Open-

vocabulary grounding with vision-language models (VLMs) enables target specification in natural language while handling visual variation [22]. Recent VLM-based models also show that semantic priors can support task-relevant reasoning and parameter extraction when coupled with executable control stacks [23]–[25], improve generalization in manufacturing and human-robot collaboration [26], and provide action-relevant structure such as articulation axes and constraints [25], [27]. Nonetheless, the key unresolved problem is converting these semantic cues into continuous, safety-constrained assistance that remains valid during online replanning, viewpoint changes, and real manipulation.

In this paper we present a shared-autonomy system for mobile manipulation that remains robust in clutter and supports both assisted teleoperation and user-triggered autonomous execution. The robot continuously maintains a local 3D collision representation from onboard perception and performs online collision-aware motion generation to enable real-time obstacle avoidance during end-effector motion. To reduce operator workload during approach and alignment, we introduce a potential-field guidance layer that continuously corrects the operator's end-effector commands toward the intended target while preserving operator authority and enforcing collision constraints. We further generalize target specification by integrating an open-vocabulary perception module that grounds the operator's query in the scene and localizes a 3D grasp frame, which is used consistently for both shared-control guidance and autonomous task execution. Fig. 1 shows the framework in operation, in which the operator commands the arm through free motion and gestures, and the assisted end-effector is drawn toward the language-grounded target during a valve manipulation task. The autonomous execution module can be explicitly triggered via hand commands, enabling reliable task completion once intent is confirmed. We evaluate the complete pipeline on real industrial tasks, including wheel-valve operation and a drill pick-and-place, demonstrating safe, collision-aware manipulation and successful contact-rich execution in clutter.

The main contributions of this paper are:

- A shared-autonomy pipeline for mobile manipulation that couples calibration-free vision-based teleoperation, open-vocabulary target grounding, and gesture-triggered autonomous execution, converting a language query into a world-latched 3D grasp frame used to carry the operator from intent decoding to task completion.
- Continuous collision-aware command tracking that integrates onboard volumetric mapping with sampling-based MPC, subjecting teleoperated and autonomous motion to the same constraints while the manipulation target is separated from the obstacle map online.
- A potential-field assistance layer that corrects the operator's reference toward the grounded target within an activation region, with the operator's command remaining the dominant term at every control cycle.

II. Methods

The proposed shared-control framework, shown in Fig. 2, decodes operator intent into safe arm motion via a modular perception-control pipeline[1]. A vision-based teleoperation interface estimates a body-centered operator frame and maps wrist and hand motion to an end-effector position reference, gripper roll command, and discrete gripper actions. An onboard volumetric mapping module reconstructs a local 3D model of the workspace from the robot's stereo sensing for collision checking, and a GPU-accelerated reactive motion generator tracks the commanded reference at interactive rates while enforcing self- and environment-collision avoidance. On this backbone, an open-vocabulary grounding module and a potential-field assistance layer correct motion toward semantically grounded targets, and a user-triggered autonomous mode completes well-defined grasps. Each module is detailed below.

A. Vision-Based Teleoperation Interface

To teleoperate the robot arm without wearables or external motion-capture infrastructure, the proposed framework uses a ZED 2i RGB-D camera to acquire synchronized color and depth images. The pipeline can be adapted to any RGB-D sensor that provides aligned depth and known camera intrinsics. Rather than relying on fiducials or an explicit user calibration stage, the system continuously estimates a body-centered reference frame directly from the operator's pose.

1) Body Tracking and Frame Estimation: The RGB data is processed by MediaPipe Pose [29] to extract 33 skeletal landmarks, shown in Fig. 2. We use the shoulders ($\mathbf{p}_{\mathrm{sh}}$), hips ($\mathbf{p}_{\mathrm{hp}}$), ankles ($\mathbf{p}_{\mathrm{ak}}$), and right wrist ($\mathbf{p}_{\mathrm{wr}}$). Each 2D landmark (u, v) is deprojected into a 3D point $\mathbf{p}_{\mathrm{cam}}$ in the camera frame using the camera intrinsics (f_x, f_y, c_x, c_y) and the median depth d from an aligned 5×5 pixel neighborhood, as defined by Eq. 1.

$$\mathbf{p}_{\mathrm{cam}} = \begin{bmatrix} (u - c_x)d/f_x \\ (v - c_y)d/f_y \\ d \end{bmatrix}. \tag{1}$$

A dynamic body-centered reference frame $\{B\}$ is then constructed from the operator's skeletal features. This approach requires no initial calibration step, as the frame is updated continuously from the detected pose, allowing free movement within the camera's field of view without degrading tracking quality. We define the centroids for the shoulders ($\mathbf{c}_{\mathrm{sh}}$), hips ($\mathbf{c}_{\mathrm{hp}}$), and ankles ($\mathbf{c}_{\mathrm{ak}}$) by averaging the left and right joint positions, as given by Eq. 2.

$$\mathbf{c}_i = \frac{\mathbf{p}_{i_R} + \mathbf{p}_{i_L}}{2}, \quad \text{for } i \in \{\mathrm{sh}, \mathrm{hp}, \mathrm{ak}\}. \tag{2}$$

To improve robustness under partial occlusion, the provisional vertical axis $\hat{\mathbf{u}}$ follows a hierarchical fallback, given by Eq. 3.

$$\hat{\mathbf{u}} = \begin{cases} \frac{\mathbf{c}_{\mathrm{sh}} - \mathbf{c}_{\mathrm{ak}}}{\|\mathbf{c}_{\mathrm{sh}} - \mathbf{c}_{\mathrm{ak}}\|} & \text{if ankles are visible,} \\ \frac{\mathbf{c}_{\mathrm{sh}} - \mathbf{c}_{\mathrm{hp}}}{\|\mathbf{c}_{\mathrm{sh}} - \mathbf{c}_{\mathrm{hp}}\|} & \text{if ankles are occluded but hips are visible,} \\ \hat{\mathbf{u}}_{\mathrm{cam}} & \text{if both hips and ankles are occluded.} \end{cases} \tag{3}$$

[1]A GitHub repository containing the code will be made available.

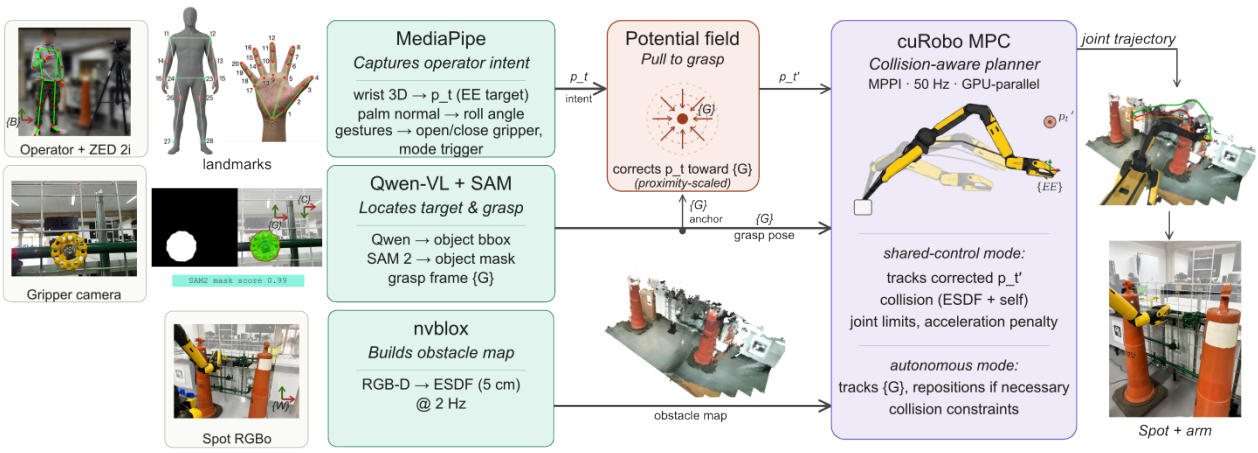

Fig. 2: **System overview.** Three perception modules supply the controller. MediaPipe converts operator wrist motion, palm orientation, and hand gestures from a ZED 2i RGB-D feed into a teleoperation reference $\mathbf{p}_t$, gripper commands, and a mode trigger. The highlighted body landmarks (shoulders, hips, ankles) define the operator frame $\{B\}$, and the palm normal from the wrist and metacarpophalangeal landmarks drive the gripper roll. Qwen3-VL [28] with SAM 2 grounds an open-vocabulary description against Spot's gripper camera, producing a grasp frame $\{G\}$ for objects such as the wheel valve illustrated. nvblox fuses Spot's stereo into a 5 cm voxel TSDF and exposes an ESDF for distance queries. A proximity-scaled potential field corrects $\mathbf{p}_t$ toward $\{G\}$, producing the modulated reference $\mathbf{p}'_t$. cuRobo's MPPI-based MPC switches between two modes. Shared-control mode tracks $\mathbf{p}'_t$ under collision, joint-limit, and smoothness costs. Autonomous mode, engaged by the gesture trigger, tracks $\{G\}$ directly to complete the task. The output joint trajectory is sent to Spot's manipulator.

The lateral axis is always defined by the shoulder span, and the remaining axes are obtained through successive cross products, as given by Eq. 4.

$$\hat{\mathbf{x}} = \frac{\mathbf{p}_{\mathrm{sh}_R} - \mathbf{p}_{\mathrm{sh}_L}}{\|\mathbf{p}_{\mathrm{sh}_R} - \mathbf{p}_{\mathrm{sh}_L}\|}, \qquad \hat{\mathbf{z}} = -\frac{\hat{\mathbf{x}} \times \hat{\mathbf{u}}}{\|\hat{\mathbf{x}} \times \hat{\mathbf{u}}\|}, \qquad \hat{\mathbf{y}} = \hat{\mathbf{x}} \times \hat{\mathbf{z}}. \tag{4}$$

The origin is placed at the torso centroid and subsequently shifted to the right-shoulder projection so that the teleoperation frame is aligned with a virtual arm base analogous to the robot's shoulder.

2) Landmark Filtering and Denoising: Depth-based skeleton tracking often suffers from transient artifacts and outliers. To mitigate these in the reconstructed 3D landmarks, we employ a threshold-based rejection filter followed by temporal smoothing. Let ΔP_t denote the raw landmark spatial displacement at frame t, v_t its instantaneous velocity, and $\Delta\theta$ the angular change of the frame axes. An update is accepted only if it satisfies spatial, kinematic, and angular constraints simultaneously, as given by Eq. 5.

$$C_t = \underbrace{\|\Delta P_t\| \le 0.40}_{\text{dist. (m)}} \wedge \underbrace{v_t \le 1.5}_{\text{vel. (m/s)}} \wedge \underbrace{\Delta\theta \le 30^\circ}_{\text{ang.}} \tag{5}$$

When C_t fails, the landmark state P_t^* holds its previous value. The accepted trajectory is then smoothed using an exponential moving average (EMA) filter, as given by Eq. 6.

$$\hat{P}_t = \alpha P_t^* + (1-\alpha)\hat{P}_{t-1}, \tag{6}$$

where the smoothing factor α is empirically tuned within the range $[0.3, 0.5]$.

3) End-Effector Command Generation: The right-wrist landmark $\mathbf{p}_{\mathrm{wr}}^{\mathrm{cam}}$, obtained from Eq. 1, is transformed into the body-centered frame $\{B\}$ using the rotation matrix $R_B = [\hat{\mathbf{x}}\ \hat{\mathbf{y}}\ \hat{\mathbf{z}}]$ from Eq. 4 and the origin $\mathbf{o}_B$. The resulting coordinate is scaled by a factor s, defined as the ratio between the robot arm's reachable workspace (approximately 1 m) and the operator's arm length, estimated online from the deprojected shoulder, elbow, and right-wrist landmarks as the sum of the upper-arm and forearm segment lengths, which are pose-invariant and therefore require no dedicated calibration step. The robot end-effector position reference is then computed as given by Eq. 7.

$$\mathbf{p}_{\mathrm{target}}^{\mathrm{robot}} = s\, R_B^\top (\mathbf{p}_{\mathrm{wr}}^{\mathrm{cam}} - \mathbf{o}_B) + \Delta\mathbf{p}_{\mathrm{sh}}, \tag{7}$$

where $\Delta\mathbf{p}_{\mathrm{sh}}$ is the constant shoulder offset of the robot arm with respect to the quadruped base.

4) Hand Pose and Gripper Commands: For hand orientation and grasp control, MediaPipe Hands [29] extracts 21 hand landmarks, as illustrated in Fig. 2. The gripper roll reference ϕ is derived from the palm normal vector $\mathbf{n}$, computed from the wrist ($\mathbf{p}_0$), the index metacarpophalangeal joint ($\mathbf{p}_5$), and the little metacarpophalangeal joint ($\mathbf{p}_{17}$), as defined by Eq. 8.

$$\mathbf{n} = \frac{(\mathbf{p}_{17} - \mathbf{p}_0) \times (\mathbf{p}_5 - \mathbf{p}_{17})}{\|(\mathbf{p}_{17} - \mathbf{p}_0) \times (\mathbf{p}_5 - \mathbf{p}_{17})\|}, \qquad \phi = \mathrm{atan2}(n_y, n_x). \tag{8}$$

A dedicated gesture-recognition module generates discrete gripper commands when *Open Palm* and *Closed Fist* gestures are detected. During these events, the roll command is briefly frozen to avoid transient orientation changes while the gripper is opening or closing. A separate finger-count gesture selects the operating mode. When the operator raises one finger, the

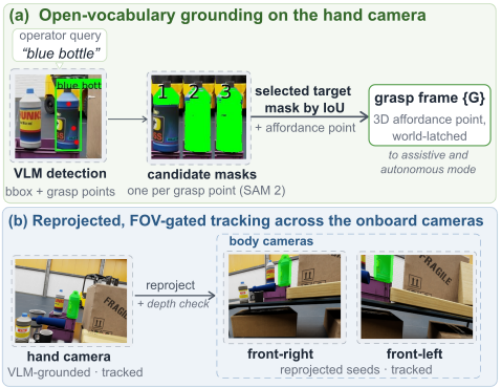

Fig. 3: Open-vocabulary grounding and multi-camera tracking. (a) A free-form operator query is grounded by Qwen3-VL on the hand camera, which returns a bounding box and candidate grasp points. Each point is segmented by SAM 2, and the affordance point is selected by the best mask-to-box IoU, defining the world-latched grasp frame $\{G\}$. (b) The single hand camera detection is reprojected, gated on field of view and depth consistency, into the two body cameras.

shared-control teleoperation mode is engaged, while raising two fingers engages autonomous execution.

B. Volumetric Mapping and Collision Representation

The motion generator requires a continuously updated 3D model of the local workspace for collision avoidance. Since the quadruped operates outside fixed workcells, this model must be built from onboard sensing and remain robust to self-occlusions introduced by the arm during execution. We use *nvblox* [20], a GPU-accelerated volumetric mapping framework, to fuse depth observations from the robot's onboard stereo cameras into a TSDF. The TSDF is built with a voxel size of $5\,\mathrm{cm}$, chosen as a compromise between geometric fidelity and real-time performance during manipulation.

a) Local workspace integration: The mapping volume is restricted to the region relevant for arm motion. The projective integration distance is limited to $1.5\,\mathrm{m}$ from the onboard cameras, removing distant background structures that are irrelevant for collision checking while preserving nearby scene geometry around the manipulation workspace, and reducing memory usage and stabilizing the local map during operation.

b) Interface to the motion generator: The volumetric map is directly used by the motion generator via the ESDF, which is maintained and published at $2\,\mathrm{Hz}$. Inside the motion generation module, the incoming ESDF enables constant-time distance queries within the *cuRobo* control loop for collision evaluation, so the controller always operates on the latest available local reconstruction even though the map refreshes at a lower rate than the control loop.

C. Open-Vocabulary Object Grounding and Segmentation

The shared-control assistance requires a semantic target, i.e., the target object the operator intends to manipulate, localized in 3D and consistently separated from the surrounding scene. The target is obtained by grounding a natural-language description in the gripper-camera image with a VLM and re-projecting the resulting segmentation to the robot's other onboard camera feeds, as illustrated in Fig. 3.

1) Language Grounding and Affordance Point: The operator specifies the target with a free-form text prompt, for example "wheel valve". Qwen3-VL processes the gripper-camera image and returns, for the queried object, a bounding box, a confidence score, and a small set of candidate grasp (affordance) points, all in normalized image coordinates. Because the gripper camera rolls with the wrist, the image is first rotated to a gravity-upright orientation using the camera-to-world rotation obtained from the kinematic tree, and the detection is mapped back to the native image. Grounding is requested on demand and gated on view stability, low gripper-camera velocity and minimal frame latency, so that the VLM operates on a well-posed image.

2) Multi-Camera Segmentation and Tracking: The target is segmented continuously in the three onboard views, namely the gripper (*hand*) camera and the two body cameras (*frontleft* and *frontright*), using a promptable video segmentation model, SAM 2 [30], with one streaming predictor per camera. Each predictor maintains a temporal memory so the mask propagates frame to frame without re-querying the VLM. In deployment, a warm grounding call takes a median of $2.2\,\mathrm{s}$, while each predictor sustains roughly $100\,\mathrm{ms}$ per inference for a mask rate of 2–$2.5\,\mathrm{Hz}$, matching the map update rate of Sec. II-B, with the $50\,\mathrm{Hz}$ controller tracking the world-latched frame in between.

On the hand camera, both the bounding box and the affordance points returned by the VLM are used. Each candidate grasp point is segmented and the affordance point is selected as the one whose mask agrees best, in intersection-over-union, with the bounding box, which yields a single point consistent with the detected object extent. To compensate for grounding latency, the selected affordance point is back-projected to a 3D point using the registered depth and camera intrinsics, then re-projected into the current frame via the kinematic tree before the streaming predictor initializes, so that the prompt lands on the object despite operator and robot motion.

The two body cameras are seeded by reprojecting the hand-tracked object position $\mathbf{p}_{\mathrm{obj}}$ into each secondary image plane (Fig. 3b), as given by Eq. 9.

$$u = f_x \frac{X}{Z} + c_x, \quad v = f_y \frac{Y}{Z} + c_y, \quad [X\ Y\ Z]^\top = R\,\mathbf{p}_{\mathrm{obj}} + \mathbf{t}, \tag{9}$$

where $(R, \mathbf{t})$ is the world-to-camera transform at the frame timestamp. This provides a positive prompt and, on first acquisition, a bounding box sized from the object's measured extent at the expected depth Z. A depth-consistency check rejects a candidate mask whose measured depth disagrees with the reprojected expectation. Each camera publishes a per-frame binary mask stamped with the exact frame from which it was computed.

3) Dynamic Object and Obstacle Separation: The segmentation masks separate the manipulation target from the collision environment within the volumetric map of Sec. II-B. For every depth image, pixels inside the mask are fused

into a *dynamic* layer, while the remaining pixels build the *static* TSDF from which the ESDF collision world is derived. The tracked object is therefore continuously excluded from the static obstacle representation. This separation is essential for collision-aware assistance, since a target fused into the static map would lead the motion generator to treat the very object the operator is reaching for as an obstacle and refuse to approach it. As a second, planning-side safeguard, the periodic ESDF query carries an axis-aligned bounding box that encloses the target, sized from the hand mask and its depth statistics, and is carved out of the map on each update, removing any residual single-frame leakage.

4) Grasp Frame: The 3D position of the affordance point defines the target grasp frame $\{G\}$, expressed in the world frame. To reject transient segmentation and depth noise under body motion, $\{G\}$ is world-latched, i.e., it holds a fixed position unless several consecutive measurements confirm that the object has genuinely moved, in which case it re-latches. This frame is the semantic target consumed by the potential-field assistance of Sec. II-E.

5) Tracking Lifecycle: The perception pipeline is governed by a lightweight state machine. The system is initialized on operator demand or when the target prompt changes: a grounding request runs the VLM, and the resulting seed places the per-camera predictors in a brief initialization window while they cold-start. Once a predictor locks onto the object, the system enters a tracking state in which the masks propagate through SAM 2 memory and the grasp frame $\{G\}$ is updated. If the mask is lost, the target is no longer marked dynamic and would otherwise fuse into the static map, so the assistance freezes $\{G\}$ at its last world-latched position and the predictors attempt to re-acquire the object from their temporal memory, which recovers tracking without re-invoking the VLM once the object re-enters view. A new VLM grounding is requested only on an explicit operator trigger or a change of target, rather than automatically on every loss, so that transient occlusions during manipulation do not restart the pipeline.

D. Collision-Aware Reactive Motion Generation

The control module converts the operator's Cartesian reference $\mathbf{p}_{\text{target}}^{\text{robot}}$ (Eq. 7) and roll command ϕ (Eq. 8) into feasible joint-space motion while continuously enforcing collision avoidance. We use *cuRobo* [31], which formulates reactive motion generation as a GPU-accelerated model predictive control (MPC) scheme. At each control cycle, the solver samples N_p candidate control sequences using a Model Predictive Path Integral (MPPI) update. The optimization runs with an integration step of $dt = 0.02\,\text{s}$, corresponding to a control frequency of $50\,\text{Hz}$, and each candidate predicts a trajectory over a horizon of H steps. For every candidate trajectory, the GPU evaluates a multi-objective cost composed of: (i) Cartesian tracking error with respect to the commanded end-effector reference, (ii) joint-limit penalties, (iii) smoothness penalties on the generated motion, and (iv) collision costs associated with both self-collision and the environment. The final control action is obtained via standard MPPI cost-weighted averaging in a receding-horizon fashion, with lower-cost trajectories contributing more strongly to the executed command. In our implementation, $N_p = 400$ and $H = 24$.

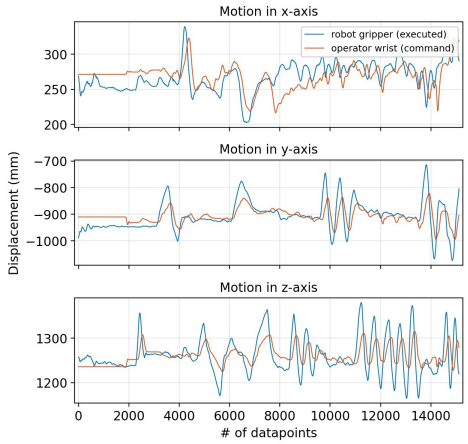

Fig. 4: Teleoperation tracking accuracy. Operator wrist (command) and robot gripper (executed) displacement along each axis, measured by a six-camera OptiTrack system. The interface achieves a positional RMSE of $59\,\text{mm}$.

The robot geometry is represented by a set of collision spheres generated from the URDF using NVIDIA Isaac Sim's *Lula Robot Description Editor* [32]. The end-effector objective is initialized relative to the current robot state, and inverse kinematics is solved locally around the current joint configuration. For environment collision checking, the controller queries the ESDF described in Sec. II-B.

E. Potential-Field Shared-Control Assistance

The teleoperation interface of Sec. II-A lets the operator command the end-effector freely, but precise alignment with a target during the final approach is demanding under camera-based control. To reduce this effort while preserving operator authority, we introduce an attractive potential field that biases the operator's reference toward the grounded grasp frame $\{G\}$ (Sec. II-C) as the end-effector nears it.

Let $\mathbf{p}_t \triangleq \mathbf{p}_{\text{target}}^{\text{robot}}$ denote the operator's commanded end-effector reference (Eq. 7) and $\mathbf{p}_G$ the position of $\{G\}$, both in the robot body frame. The potential-field assistance engages only within an activation radius $d_a = 0.4\,\text{m}$ of the target, following Eq. 10.

$$\mathbf{p}'_t = \begin{cases} \mathbf{p}_t + k_{\text{att}}\,(\mathbf{p}_G - \mathbf{p}_t), & \|\mathbf{p}_G - \mathbf{p}_t\| \le d_a, \\ \mathbf{p}_t, & \text{otherwise}, \end{cases} \tag{10}$$

where $k_{\text{att}} \in (0, 1)$ sets the strength of the correction and $\mathbf{p}'_t$ is the modulated reference tracked by the motion generator. We set $k_{\text{att}} = 0.2$. Only the attractive component is used, since repulsion is delegated to the collision costs of Sec. II-D.

Since $k_{\text{att}} < 1$, the operator's command remains the dominant term and authority is retained. The operator chooses when to approach and can always pull away, while the field only removes the fine residual error of the final centimeters. The distance is measured from the operator's reference rather than the current end-effector, so the assist engages based on

where the operator is aiming. Because the correction is applied at every control cycle, the tracked reference is drawn smoothly toward $\{G\}$ as long as the operator holds the end-effector within d_a, producing a stable and progressively tightening approach, and since $\mathbf{p}'_t$ is tracked by the collision-aware generator of Sec. II-D, the assisted motion remains subject to the same collision constraints.

F. User-Triggered Autonomous Execution

The shared-control assistance keeps the operator engaged throughout the approach. For well-defined or repetitive grasps, however, the operator can delegate the final approach and grasp to the robot once intent is confirmed. Following the gesture protocol of Sec. II-A, the operator raises two fingers to enter autonomous mode once the target has been grounded, and then initiates execution with a *Closed Fist*, so autonomy engages only when the operator chooses to hand off. As in shared-control mode, an *Open Palm* commands the gripper to release.

Autonomous execution reuses the pipeline rather than introducing a parallel one. The goal is the same grasp frame $\{G\}$ produced by the grounding module (Sec. II-C), with no separate detector, fiducial, or hand-tuned target. The end-effector is brought to $\{G\}$ by the same collision-aware reactive motion generator of Sec. II-D, the only difference being that the tracked reference is $\{G\}$ rather than the operator's modulated command $\mathbf{p}'_t$. The autonomous approach therefore inherits the identical self- and environment-collision guarantees enforced during teleoperation, and the target remains carved out of the static obstacle map (Sec. II-C) so the controller can reach it without treating it as an obstacle.

Once the end-effector is aligned with $\{G\}$, the final grasp is delegated to Spot's onboard grasp service [33], seeded with the grounded target point. The perception and shared-control layers decide *what* and *where* to grasp, while the hardware-calibrated grasp primitive resolves the last-centimeter wrist alignment and closure, producing robust, repeatable grasps without reimplementing low-level grasp control. If the grounded target lies outside the current arm workspace, the grasp service can reposition the base to bring the object within reach, gated on operator confirmation and using Spot's onboard locomotion obstacle avoidance.

Throughout autonomous execution, the operator retains high-level authority and can intervene at any time, resuming shared-control teleoperation to complete contact-rich steps such as turning the grasped valve.

III. Experiments and Results

A. Experimental Setup

All experiments were performed on a Boston Dynamics Spot quadruped robot equipped with the 6-DoF Spot Arm and its jaw gripper. The operator was tracked by a ZED 2i RGB-D camera placed in front of the operator workspace. Perception, grounding, mapping, and control ran on an offboard workstation with an NVIDIA RTX A2000 GPU (12 GB) and an Intel Xeon CPU, communicating with the robot over Ethernet. The robot's base remained stationary during manipulation.

The framework was validated in three stages: the positional accuracy of the teleoperation interface against motion-capture ground truth, collision-aware shared control in a cluttered workspace, and the complete pipeline on two contact-rich manipulation tasks. Each manipulation task was executed under five conditions, pure teleoperation, teleoperation with only collision avoidance, teleoperation with only potential-field assistance, the full shared-control framework, and user-triggered autonomous execution, with five trials per condition performed by an operator. A trial is successful if the object is grasped and the task completed, and any contact with the surrounding structure ends the trial as a failure. The framework was further completed in all trials by a second user[2].

B. Teleoperation Tracking Accuracy

To quantify how faithfully the interface transfers operator motion to the robot, reflective markers were attached to the operator's wrist and the Spot Arm's wrist, and both were tracked by an OptiTrack system comprising six PrimeX 41 cameras, which provides sub-millimeter 3D accuracy and which we employed as ground truth. The operator performed unscripted free motions across the workspace while the robot tracked the commanded reference through the full pipeline of Sec. II-A and Sec. II-D. The positional error is computed between the operator's wrist trajectory, mapped into the robot workspace through the scale factor of Eq. 7, and the robot end-effector trajectory, both measured by the OptiTrack system. The reported error therefore reflects the complete pipeline, including perception, filtering, scaling, and motion generation, rather than the tracking accuracy of the controller alone.

Fig. 4 shows the wrist and gripper motion along each axis. The interface achieved a positional RMSE of 59 mm (21 mm in x, 42 mm in y, and 35 mm in z), comparable to prior camera-based interfaces for legged manipulators [34], [35], and the residual error is precisely what the assistance layer absorbs during the final approach. The gripper closely follows the commanded motion across all axes, including the fast direction reversals in the second half of the trial. The largest transient deviations coincide with these rapid reversals, where the smoothing filter and the finite tracking bandwidth of the arm introduce lag, while during slow, deliberate motion the executed trajectory stays close to the command. The error therefore behaves as lag rather than drift, and the interface remains accurate in the regime where teleoperation precision is paramount, the final approach to a target.

C. Collision-Aware Shared Control

The second experiment demonstrates the collision avoidance of Sec. II-D during teleoperation in clutter. The workspace, shown in Fig. 1b, consists of a manipulation panel with industrial piping, valves, and a pressure gauge, flanked by traffic cones that constrain the free space around the arm. The operator commanded lateral end-effector motions across the workspace, deliberately sweeping toward and into the cones, while the signed distance of the commanded reference and

[2]A demonstration video of the experiments is available at Google Drive.

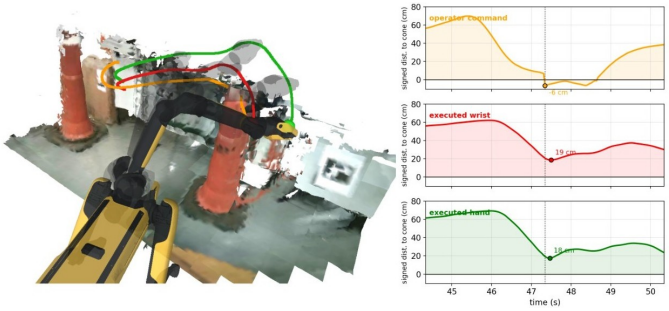

Fig. 5: Collision-aware shared control. Left: reconstructed workspace with the commanded (orange), executed wrist (red), and executed hand (green) trajectories. Right: signed distance of each to the nearest cone over time. The operator deliberately drives the command 6 cm into the obstacle, while the executed wrist and hand never come closer than 19 cm and 18 cm.

of the executed arm links to the obstacles was recorded. Figure 5 shows a representative sweep, with the reconstructed workspace, the commanded and executed trajectories, and the signed distance of each to the nearest cone over time. The commanded trajectory penetrates the cone surface, reaching -6 cm at its deepest. The executed motion, in contrast, never approaches contact, where the wrist keeps a minimum distance of 19 cm and the hand 18 cm, as the MPPI collision costs correct the tracked path around the obstacle while continuing to follow the lateral component of the command. The deviation is confined to the vicinity of the cone, and tracking resumes as soon as the command returns to free space. The observed margin exceeds the collision-cost activation distance because the arm geometry is represented by collision spheres and the MPPI horizon penalizes approaching trajectories before the activation distance is reached, with distances reported to the wrist and hand frames rather than to the sphere surfaces.

D. Contact-Rich Manipulation Tasks

The final experiment evaluates the complete framework on two tasks under the five conditions of Sec. III-A. In the assisted conditions the operator performs the approach and grasp, with the potential field active, while in the autonomous condition the operator delegates the approach and grasp to the robot (Sec. II-F). In all conditions the target is specified by the same free-form text prompt and grounded as in Sec. II-C.

1) Valve Manipulation: The first task uses the industrial panel of Fig. 1b. The operator grounds the target with the prompt "wheel valve", approaches, grasps the valve, and rotates it by a quarter turn. In the autonomous condition, the operator raises two fingers and confirms with a Closed Fist, after which the robot completes the approach and grasp on $\{G\}$, and in every condition the quarter-turn rotation is performed by the operator under shared control, consistent with Sec. II-F. Fig. 1c–e shows the assisted approach, plotting the operator's commanded reference, the assisted end-effector trajectory on which the field acts, and the gripper tip, all relative to $\{G\}$. Outside the 0.4 m activation radius the trajectories follow each other, since the field is inactive. Once the command enters the activation region, the assisted trajectory is drawn toward $\{G\}$, and the gripper tip converges onto the affordance point. The trailing loops correspond to the quarter-turn rotation performed by the operator after the grasp, executed under the same assistance. The commanded motion was amplified by the workspace scale factor s of Eq. 7, estimated online from the operator's arm length.

2) Pick and Place: The second task is a pick-and-place of a power drill resting on a box. The operator grounds the target with the prompt "drill", grasps it, teleoperates its transport, and releases it with an Open Palm. This task complements the valve by requiring a free-object grasp, transport under load, and placement, rather than a fixed-fixture interaction.

E. Ablation Study

Table I summarizes the outcomes of the ablation studies. Without either module, the limited depth perception of the camera interface makes the tasks largely infeasible, as the operator either contacts the surrounding structure or cannot align the gripper with the target. With collision avoidance alone, no contact occurs in any trial, but precise final alignment through the gripper camera remains difficult, and most trials fail at the grasp itself. With the potential field alone, the missed grasps largely disappear, but fast motions and fine corrections near the structure produce contacts that the assistive field cannot prevent. The two modules, therefore, suppress complementary failure modes, and with both active, all trials succeed. The autonomous condition failed once per task due to inaccurate depth estimates in the grasp service, which led to a misplaced final grasp pose, and the operator recovered by resuming shared control. In every trial, the grounding module localized the target from the free-form prompt, and the same pipeline handled a fixed, environment-mounted target and a free, movable object without retuning.

TABLE I: Task success across the ablation conditions, five trials per condition. C denotes collision avoidance, and PF denotes the potential-field assistance.

Condition	Valve	Pick and place
Teleoperation (no C, no PF)	0/5	1/5
Teleoperation + C	2/5	2/5
Teleoperation + PF	3/5	4/5
Full framework (C + PF)	**5/5**	**5/5**
Autonomous execution	4/5	4/5

IV. Conclusion

This paper presented a shared-autonomy framework for mobile manipulation that couples calibration-free vision-based teleoperation with open-vocabulary target grounding, continuous collision-aware motion generation, potential-field assistance, and a user-triggered autonomous execution mode, on a quadruped manipulator. The operator's motion is captured without wearables, fiducials, or a calibration stage. The intended target is specified by a free-form text prompt and tracked across the robot's onboard cameras, and every commanded motion, assisted or autonomous, is executed through

the same collision-aware motion generator. The framework was validated on a Boston Dynamics Spot in three stages. The teleoperation interface achieved a positional RMSE of 59 mm against motion-capture ground truth. In a cluttered workspace, the motion generator kept the arm at least 18 cm from the obstacles while the operator deliberately commanded the reference 6 cm into them. In an industrial valve manipulation and a pick-and-place task, the full framework succeeded in all trials, while every ablated condition produced failures, through contact when the collision costs were removed and through missed grasps when the assistance was removed, showing that the two layers suppress complementary failure modes.

The current validation has limitations that define the next steps. The evaluation involved two operators and no external interface baseline, so a user study with more participants, comparisons against joystick and wearable-based teleoperation, and workload measures will be conducted in future work. Finally, the grounded grasp frame is a single affordance point, and richer geometric attributes, such as manipulation axes inferred from interaction rather than perception, would broaden the range of tasks the autonomous mode can complete without operator involvement.

REFERENCES

[1] M. Ramezani, M. Brandao, B. Casseau, I. Havoutis, and M. Fallon, "Legged robots for autonomous inspection and monitoring of offshore assets," in *Offshore Technology Conference*. OTC, 2020.

[2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[3] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, 2022.

[4] M. S. Lopes, A. P. Moreira, M. F. Silva, and F. Santos, "A review on quadruped manipulators," in *EPIA Conference on Artificial Intelligence*. Springer, 2023, pp. 199–211.

[5] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, p. eadg5014, 2023.

[6] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning whole-body manipulation for quadrupedal robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, 2024.

[7] A. Rigo, M. Hu, S. K. Gupta, and Q. Nguyen, "Hierarchical optimization-based control for whole-body loco-manipulation of heavy objects," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 322–15 328.

[8] M. Rubagotti, T. Taunyazov, B. Omarali, and A. Shintemirov, "Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control," *IEEE Robotics and Automation Letters*, 2019.

[9] C. Cruz Ulloa, D. Domínguez, J. del Cerro, and A. Barrientos, "Analysis of mr–vr tele-operation methods for legged-manipulator robots," *Virtual Reality*, vol. 28, no. 3, p. 131, 2024.

[10] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015, pp. 526–531.

[11] C. Zhou, Y. Wan, C. Peers, A. M. Delfaki, and D. Kanoulas, "Advancing teleoperation for legged manipulation with wearable motion capture," *Frontiers in Robotics and AI*, vol. 11, p. 1430842, 2024.

[12] E. Krotkov, M. Diftler, R. Ambrose, A. Deguet, B. Pires, M. DeDonato, Z. Kazi, and W. Kim, "The darpa robotics challenge finals: Results and perspectives," *Journal of Field Robotics*, 2017.

[13] S. Kawatsuma, M. Fukushima, and T. Okada, "Emergency response by robots to fukushima-daiichi accident: summary and lessons learned," *Industrial Robot: An International Journal*, 2012.

[14] B. Güleçyüz, R. Balachandran, M. Panzirsch, H. Singh, T. Hulin, X. Xu, and E. Steinbach, "Enhancing shared autonomy in teleoperation under network delay: Transparency-and confidence-aware arbitration," *IEEE Robotics and Automation Letters*, 2025.

[15] I. Ozdamar, M. Laghi, G. Grioli, A. Ajoudani, M. G. Catalano, and A. Bicchi, "A shared autonomy reconfigurable control framework for telemanipulation of multi-arm systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9937–9944, 2022.

[16] M. Hagenow, E. Senft, R. Radwin, M. Gleicher, B. Mutlu, and M. Zinn, "Corrective shared autonomy for addressing task variability," *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 3720–3727, 2021.

[17] A. Phung, G. Billings, A. F. Daniele, M. R. Walter, and R. Camilli, "A shared autonomy system for precise and efficient remote underwater manipulation," *IEEE Transactions on Robotics*, 2024.

[18] B. Guan, R. V. Godoy, F. Sanches, A. Dwivedi, and M. Liarokapis, "On semi-autonomous robotic telemanipulation employing electromyography based motion decoding and potential fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[19] R. V. Godoy, B. Guan, A. Dwivedi, and M. Liarokapis, "An affordances and electromyography based telemanipulation framework for control of robotic arm-hand systems," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6998–7004.

[20] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2698–2705.

[21] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "iSDF: Real-time neural signed distance fields for robot perception," in *Robotics: Science and Systems (RSS)*, 2022.

[22] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*. Springer, 2024, pp. 38–55.

[23] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: an embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23, 2023.

[24] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[25] S. Huang, H. Chang, Y. Liu, Y. Zhu, H. Dong, A. Boularias, P. Gao, and H. Li, "A3vlm: Actionable articulation-aware vision language model," in *Conference on Robot Learning*. PMLR, 2025, pp. 1675–1690.

[26] J. Fan, Y. Yin, T. Wang, W. Dong, P. Zheng, and L. Wang, "Vision-language model-based human-robot collaboration for smart manufacturing: A state-of-the-art survey," *Frontiers of Engineering Management*, vol. 12, no. 1, pp. 177–200, 2025.

[27] R. Buchanan, A. Röfer, J. Moura, A. Valada, and S. Vijayakumar, "Online estimation and manipulation of articulated objects," *Autonomous Robots*, 2026.

[28] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[29] Google, "MediaPipe: Cross-platform, customizable ML solutions for live and streaming media," https://developers.google.com/mediapipe, 2024, accessed: 2026-03-12.

[30] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 28 085–28 128.

[31] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8112–8119.

[32] NVIDIA, "Isaac sim Lula robot description editor," https://docs.isaacsim.omniverse.nvidia.com/5.1.0/manipulators/manipulators_robot_description_editor.html, 2024, accessed: 2026-03-18.

[33] Boston Dynamics, "Spot SDK: Software development kit for the spot robot," https://github.com/boston-dynamics/spot-sdk, 2025, accessed: 2025-05-31.

[34] M. V. da Silva, M. H. Carvalho, J. Negri, T. Segreto, G. J. G. Lahr, R. V. Godoy, and M. Becker, "A vision-based shared-control teleoperation scheme for controlling the robotic arm of a four-legged robot," in *2025 Latin American Robotics Symposium (LARS)*, 2025.

[35] L. A. Zick, D. Martinelli, A. Schneider de Oliveira, and V. Cremer Kalempa, "Teleoperation system for multiple robots with intuitive hand recognition interface," *Scientific reports*, 2024.