IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

LM-Aided Assistive Robot for Single-Operator Bimanual Teleoperation

Haolin Fei[1], Songlin Ma[1], Guanglong Du[2], Elmira Yadollahi[1], Hak-Keung Lam[3],
Angela Faragasso[4], Allahyar Montazeri[1], and Ziwei Wang[1]

Abstract— **Bimanual teleoperation tasks are highly demanding for human operators, requiring the simultaneous control of two robotic arms while managing complex coordination and cognitive load. Current approaches to this challenge often rely on rigid control schemes or task-specific automation that do not adapt well to dynamic environments or varied operator needs. This paper presents a novel large language model (LLM)-aided bimanual teleoperation assistant (BTLA) that helps operators control dual-arm robots through an intuitive voice command interface and variable autonomy. The BTLA system enables a hybrid control paradigm by combining natural language interaction for an assistive robot arm with direct teleoperation of the dominant robotic arm. Our system implements six core manipulation skills with varying autonomy, ranging from direct mirroring to autonomous object manipulation. The BTLA leverages the LLM to interpret natural language commands and select an appropriate assistance mode based on task requirements and operator preferences. Experimental validation on bimanual object manipulation tasks demonstrates that the BTLA system yields a 240.8% increase in success rate over solo teleoperation and a 69.9% increase over dyadic teleoperation, while significantly reducing operator mental workload. In addition, we validate our approach on a physical dual-arm UR3e robot system, achieving a 90% success rate on challenging soft-bottle handling and box-transportation tasks.**

I. Introduction

Teleoperation has become a crucial approach for operating robotic systems in environments that are inaccessible or hazardous to humans, ensuring both efficiency and safety. It has been successfully applied in diverse domains, including space rendezvous and docking [1], underwater operations [2], and remote surgery [3]. Within these contexts, dual-arm teleoperation has gained particular importance because it provides additional dexterity and more degrees of freedom compared to single-arm configurations. This capability enables the execution of more sophisticated tasks that demand precise coordination.

This work was partly supported by the Royal Society Grant (IES/R2/232291), the UK-RAS Network+, the EPSRC (Grant number: EP/Y010523/1), and the European Commission Horizon Grant Up-Skill (RIA 101070666). Songlin Ma is co-funded by the Robotics Core Science Theme under the National Nuclear Laboratory Science & Technology Programme and Nuclear Decommissioning Authority PhD Bursary. (*Corresponding author: Ziwei Wang*)

[1]Haolin Fei, Songlin Ma, Allahyar Montazeri, and Ziwei Wang are with the School of Engineering, Lancaster University, LA1 4YW, United Kingdom. (e-mail: h.fei1@lancaster.ac.uk, z.wang82@lancaster.ac.uk). [2]Guanglong Du is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China. [3]Elmira Yadollahi is with the Department of Computing and Communications, Lancaster University, LA1 4YW, United Kingdom.[4]Hak-Keung Lam is with the Department of Engineering, King's College London, Strand, London, WC2R 2LS, United Kingdom.[5]Angela Faragasso is with the FingerVision Co., Ltd., 2-4-14 Toyo, Koto-ku, Tokyo 135-0016, Japan.

Two primary modes of interaction dominate dual-arm teleoperation: (i) solo teleoperation, where a single operator controls both arms simultaneously, and (ii) dyadic teleoperation, in which two individuals each control one arm with their dominant hand. Solo teleoperation is highly sensitive to ergonomic factors, task complexity, and operator workload [4]. The operator needs to simultaneously manage the motion and coordination of two robotic arms, which can lead to increased mental workload and reduced performance [5]. In dyadic arrangements, although collaboration distributes the control effort, synchronization, communication, and arbitration between operators often limit efficiency. To mitigate these challenges, the operator can concentrate on a subset of task elements, while the robotic partner manages the complementary aspects [6]. Nevertheless, existing assistance strategies are generally either task-specific or employ fixed autonomy levels, restricting their ability to adapt to diverse environments and operator preferences. They may also require specialized training or rely on rigid command structures that do not align naturally with human communication styles.

In this paper, we propose the Bimanual Teleoperation LLM Assistant (BTLA), a system that integrates natural language communication with variable autonomy to support single operators in dual-arm teleoperation. With BTLA, the operator directly controls one robotic arm, while the secondary arm is commanded via voice instructions. An LLM interprets these instructions and selects suitable assistance modes from a predefined set of manipulation skills. The main contributions of this work are summarized as follows:

1) We introduce a flexible assistance system that reduces operator cognitive load by enabling natural language control of a secondary robotic arm without compromising task success.
2) We integrate LLM for robust natural language understanding in robotic teleoperation, enabling operators to issue complex manipulation commands intuitively.
3) We implement and validate six manipulation skills with adjustable autonomy levels (from direct mirroring to autonomous object manipulation), supporting seamless transitions between assistance modes.
4) We provide an extensive experimental evaluation that demonstrates clear improvements in task success rates and reductions in operator workload compared with conventional solo and dyadic teleoperation.

II. Related Work

Dual-arm teleoperation takes two forms: single-person bimanual (SPB), where one operator drives both arms, and dyadic, where two operators each control one arm. SPB imposes high cognitive demands as complexity grows [7], while dyadic control introduces synchronization overhead and arbitration costs that can offset the benefits of shared effort [8], [9]. Adaptive architectures that switch between controllers at runtime have been proposed to address these limitations [10].

Intuitive interfaces — gesture-based [3], VR-based [11], and haptic device-based [12] — reduce mental effort and improve situational awareness, while haptic feedback laws convey contact information to the operator [13]. On the assistance side, shared-control methods combine human intent with autonomous behaviors [14]; recent work shows that appropriate autonomy allocation reduces cognitive load by 34% [15] and decreases NASA-TLX scores across all dimensions [16].

Large language models (LLMs) and VLMs have been used for high-level robot task planning [17], [18], but purely LLM-planned behaviors remain suboptimal for tight physical interaction in dynamic environments [19]. This paper repositions the LLM not as a planner but as a human-robot interface, exploiting its natural language processing strengths to convey operator intent. Unlike hand-written grammar pipelines, the LLM handles diverse phrasings, maintains conversational context, and tolerates ambiguity — directly benefiting high-cognitive-load teleoperation.

III. Methodology

The bimanual teleoperation problem is formulated in Section III-A. Subsequently, we detail the methodology by which BTLA employs an LLM to facilitate bimanual teleoperation tasks.

A. Problem Formulation

We target SPB teleoperation in which the human directly commands a dominant arm and issues natural-language requests that drive an assistive arm. At time t, the system processes multiple input streams to determine appropriate assistance behaviors, including natural language commands l that specify desired assistive behaviors, proprioception for the dominant ($\mathbf{s}_{m,t}$) and the assistant arms ($\mathbf{s}_{a,t}$), environmental observations ($\mathbf{o}_{env,t}$), human control inputs ($\mathbf{u}_t$), and sensor measurements ($\mathbf{z}_t$). Note that l can be long-horizon, context-aware, or ambiguously defined (e.g., "move slightly upwards"), necessitating sophisticated contextual and semantic interpretation.

We assume a skill base $\mathcal{S}$ containing reusable low-level manipulation primitives. The problem formulation can be summarized as follows: given instruction l and the multi-modal state $\{\mathbf{s}_{a,t}, \mathbf{s}_{m,t}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{o}_{env,t}\}$, the embodied AI system should (1) decompose the high-level instruction l into a sequence of skills from $\mathcal{S}$ with parameters, and (2) map those skills to a control policy π for the assistant arm: $\pi = \mathrm{BTLA}(\text{skill sequence}, \text{params}, \mathbf{u}_t, \mathbf{z}_t)$. To this end, the skill knowledge in the base $\mathcal{S}$ can be adapted to accommodate different task requirements. Therefore, our focus is not on learning the primitives themselves but on selecting and instantiating the right skills online, and therefore the assistant can reliably support the human's ongoing bimanual action.

B. BTLA System Implementation

BTLA fuses natural-language understanding with robot control so that an operator can speak to the assistant arm while manually teleoperating the dominant arm. An LLM parses utterances, maintains conversational context, and returns the appropriate skill and arguments to execute. Compared to traditional template-based NLP methods, which require extensive rule engineering and struggle to handle the linguistic variations common in high-stress teleoperation scenarios, our LLM-based approach offers two critical advantages. First, the LLM maintains conversation history, enabling operators to use pronouns and references (e.g., "move it closer", "stop that") without explicit object specification. Secondly, the system interprets diverse phrasings of identical commands (tested with 47 command variations, achieving 89% consistency). These capabilities directly address operator cognitive load by eliminating the need for memorization of rigid command syntax during complex bimanual coordination.

BTLA's LLM module comprises three cooperating parts: (i) the natural language interface uses OpenAI's Whisper model for speech-to-text conversion and LLM processing to interpret operator intentions; (ii) a skill layer implementing six core primitives: *Follow*$(\cdot)$, *Symmetrical Follow*$(\cdot)$, *Approach*$(\cdot)$, *Move*$(\cdot)$, *Handover*$(\cdot)$, and *Fetch*$(\cdot)$; (iii) a policy generator that turns selected skills plus state into safe, executable robot commands, with guardrails for transitions and safety constraints. Unlike a simple skill switcher, the LLM can interpret complex instructions, understand context, and provide feedback when needed. This flexibility enables the robot assistant to adapt to a wider range of scenarios and user needs, embodying the variable autonomy principle of the BTLA framework.

Fig. 1 presents BTLA as three principal elements: the human operator, the human-robot interface, and the teleoperation environment. The operator maintains task focus by perceiving the scene through visual feedback, directly teleoperating one arm via the interface, and delegating complementary subtasks to an AI-assisted partner arm for cooperative execution. The AI-assisted arm accepts natural-language commands, retrieves the most pertinent skill from the library $\mathcal{S}$, and instantiates the required task parameters. Combining the chosen skill with environmental perception (e.g., vision and force/torque sensing), proprioceptive signals, and operator inputs yields the control policy that drives the AI-assisted arm. The LLM processes this multi-modal context to select appropriate skills. In this arrangement, the operator supplies high-level intent and oversight, while the AI-assisted arm contributes contextual understanding and the capability to achieve the objective efficiently within the teleoperation workspace. Algorithm 1 summarizes the core control loop: voice commands are parsed by the LLM to choose and invoke skills, accommodating both continuous real-time skills that persist until canceled (e.g., following)

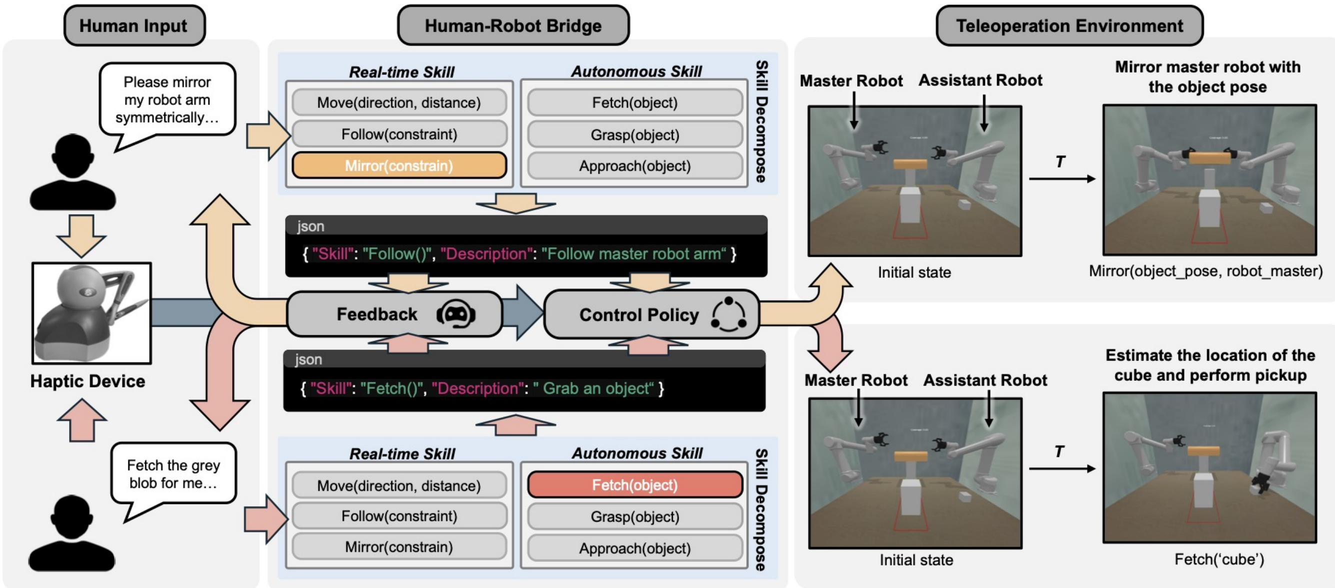

Fig. 1. Architecture diagram of the BTLA framework.

and autonomous skills that complete discrete goals (e.g., object fetching).

Explicit entry conditions and termination tests specify each skill. Real-time skills, such as *Follow*$(\cdot)$ and *Symmetrical Follow*$(\cdot)$, continuously adapt to the motions of the dominant robotic arm, whereas autonomous skills (e.g., *Fetch*$(\cdot)$ and *Handover*$(\cdot)$) execute closed-loop manipulation routines to complete discrete objectives. Intent parsing runs concurrently with action execution. Upon receiving a voice command, the controller keeps the current behavior active while the LLM pipeline interprets the new instruction, enabling smooth transitions between assistance modes. The operator may speak new commands at any time; the system completes the current motion action before switching to the requested behavior. For safety, BTLA arbitrates simultaneous voice and direct teleoperation inputs via a priority-driven state machine. Emergency directives (e.g., "stop") pre-empt all other activity and halt execution immediately; non-urgent requests are deferred until the ongoing operation finishes. During tightly coordinated phases, overlapping control channels between the two arms are temporarily lock-protected to prevent contention. A hysteresis band on state transitions suppresses rapid oscillation when contradictory commands arrive in quick succession.

Extending the skill library and task taxonomy, the framework adds a fault-handling layer that addresses command misinterpretation and kinematic singularities. It detects such conditions and initiates safe recovery or clarification routines to maintain reliable operation. To ensure safe and effective operation, BTLA implements a confirmation process before executing any task. When the robot receives a command, it first interprets the instruction and generates an execution plan. Before proceeding, it communicates this plan back to the human operator for confirmation. This step allows the operator to verify that the robot has correctly understood the command and provides an opportunity to make corrections if needed. In cases where the robot encounters singularities or potential issues during task execution, it immediately halts the operation and seeks guidance from the human operator. This interactive loop between the human operator and the robot assistant ensures a robust and adaptable system that can recover from misunderstandings and navigate complex scenarios, maintaining the balance between autonomous operation and human oversight.

Algorithm 1 Embodied AI-Assisted Robot Arm Control

```
Require: Initial skills base S with predefined skills, LLM
    initial language description l
Initialize t ← 0, skill ← None
while not finished do
    if voice commands received then
        skill ← LLM (voice commands)
        π ← BTLA (skill, skill parameters, u_t, z_t)
        if skill is real-time then
            repeat
                Execute π
                t ← t + 1
            until voice commands to stop
        else if skill is autonomous then
            repeat
                Execute π
                t ← t + 1
            until skill is done
        end if
    end if
end while
```

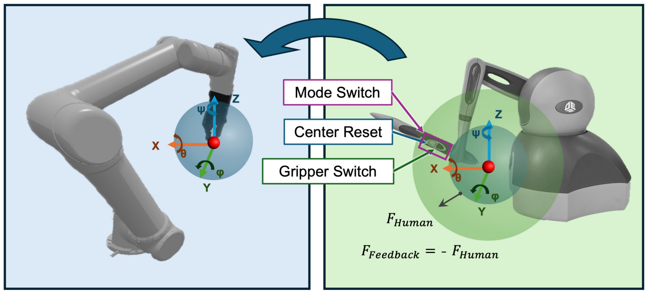

Fig. 2. Teleoperation control scheme.

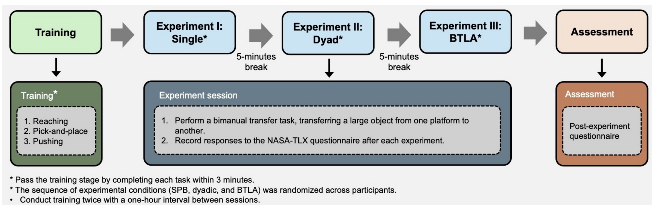

Fig. 3. Experimental procedure for assessing bimanual teleoperation. Participants first received instruction and practice on three tasks: reaching, pick-and-place, and pushing. They then completed three teleoperation sessions (SPB, Dyadic, and BTLA). These were administered in a randomized sequence unique to each participant.

C. Haptic Control

Haptic stylus displacement is mapped to end-effector velocity via $V_{i,robot} = k_v \cdot d_{i,hand}$, where k_v is a velocity gain (Fig. 2). Force feedback is rendered as $F_{i,Feedback} = k_f \cdot d_{i,hand} + F_{initial}$, providing boundary sensation and speed awareness proportional to stylus deflection. Two buttons on the stylus handle gripper toggle and origin reset; pressing both simultaneously switches between position and orientation control modes.

IV. EXPERIMENT

To examine BTLA's practical effectiveness, we conducted a series of bimanual object-manipulation experiments to evaluate:

1) How does BTLA impact task performance compared to traditional single-operator and dyadic teleoperation approaches?
2) To what extent does BTLA reduce operator workload during complex bimanual tasks?
3) What is the user experience and acceptance of LLM-based assistance in teleoperation?

To pursue these questions, we further conducted a user study with a dual-arm robotic platform performing manipulation and transport of bulky, high-mass objects. The end-to-end workflow, from operator familiarization to quantitative performance assessment, is shown in Fig. 3. We benchmark BTLA against conventional teleoperation by measuring task efficiency, operator workload, and user satisfaction.

A. Experimental Setup

1) Equipment and Software: We used two 3D Systems Touch haptic devices for operator input. Environments, robot control, and visualization were implemented with PyBullet. Custom objects were modeled in Fusion 360 and exported as URDFs to control appearance and physics consistently across trials. GPT-3.5-Turbo served as the default LLM, selected after comparison trials with GPT-4 and Mistral-7B-OpenOrca: all three exhibited comparable command-interpretation accuracy, but GPT-3.5-Turbo delivered noticeably lower response latency.

2) Initial LLM configuration: We initialize the LLM with a concise role description and response schema that clarify the robot assistant's remit and objectives. This approach eliminates the need for pre-interaction overhead, immediately turning user utterances into an executable "skill + parameters" record. The initial prompt instructs the LLM to translate spoken operator commands into a JSON-formatted script: *"Skill": "Write the function here.", "Description": Include a necessary description about this skill, as if you are talking to the user directly*. The robot assistant is equipped with the admissible manipulation skills from the skill library $\mathcal{S}$, enabling direct matching between utterances and callable functions. The LLM is programmed to provide user feedback on its actions through the "*Description*" field in the JSON script. If a command maps to a known skill, the LLM returns the corresponding script; if not, it produces a script with an empty function and a description explicitly noting that no action will be taken. This catalog- and schema-based priming (see Fig. 4) standardizes interpretation and selection of skills. Before execution, the assistant immediately confirms its understanding in natural language, yielding more fluent, reliable interactions for bimanual handling tasks.

LLM Initial Prompt Example

You are an AI assistant helping a robot arm execute missions.
Generate a script using the given format based on the user's command:
Script: {
"Skill": "Write the function here.",
"Description": "Include a necessary description about this skill, as if you are talking to the user directly. Use 'you' to address the user."
}
Available skill functions:
Follow() - Follow the master robot arm (e.g. for pushing together);
SymmetricalFollow() - Act a mirror behavior of the master robot arm;
Approach() - Move the arm to approach an object (e.g. for lifting objects together); Move(distance, direction) - Move the arm (ask user for distance in meters and direction:
"+x", "-x", "+y", "-y", "+z", "-z");
Handover() - Handover an object to the master arm ;
Fetch() - Grab an object and bring it to the master arm;
Stop() –Stop the current behavior or movement;
NoFunction() - Not a recognized command.

Give user feedback on your action in the description. If the command matches a skill, generate the corresponding script. If the command doesn't match any skills, generate a script with NoFunction() and description on not doing anything.

Always respond with a JSON script in the given format.

Fig. 4. LLM initial prompt: textual description of the mission and skills.

We implemented a keyword-based baseline system using predefined command templates (e.g., "robot follow now", "grab object red") for comparison. This system utilized rule-based parsing without LLM interpretation, representing conventional voice control approaches.

3) Skills: We distinguish two skill classes: autonomous and real-time. Autonomous skills execute a predefined sequence and terminate upon completion, such as *Handover*$(\cdot)$: transfer an object to the dominant arm; *Approach*$(\cdot)$: move the arm to an object's vicinity, e.g., to co-locate items; *Fetch*$(\cdot)$: grasp an object and deliver it to the dominant arm. Real-time skills generate continuous motion and remain

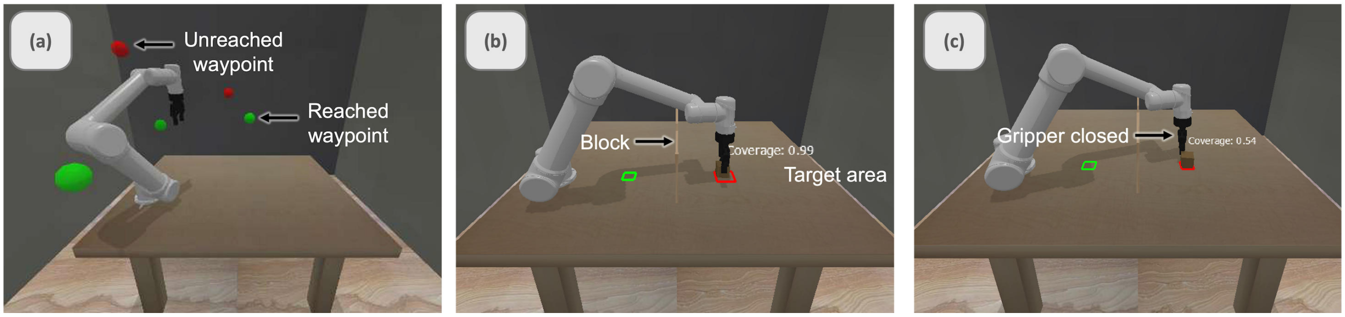

Fig. 5. Single-arm training tasks: (a) waypoint reaching; (b) obstacle-constrained pick-and-place; and (c) planar pushing to a target region.

active until a stop command is issued, such as *Follow*$(\cdot)$: track the dominant arm for cooperative pushing; *Symmetrical Follow*$(\cdot)$: mirror the dominant arm; *Move*(distance, direction): translate the arm after querying the operator for a distance in meters and a direction from "$+x$", "$-x$", "$+y$", "$-y$", "$+z$", "$-z$"). Each skill performs parameter validation and safety gating before execution.

B. Training Protocol

To standardize operator proficiency, participants completed three single-arm tasks before bimanual trials (see Fig. 5): (i) waypoint reaching through free space, (ii) obstacle-constrained pick-and-place with a 4-minute limit, and (iii) planar pushing to a target region with a 3-minute limit. This sequence progresses from free-space motion to contact-rich manipulation, ensuring fundamental control skills before coordinated bimanual actions.

C. Experimental Procedure

Ten participants (7 males, 3 females; aged 22-35) completed all conditions following ethics approval (Lancaster University FST-2024-4525-RECR-4). After training, subjects participated in randomized trials of SPB, Dyadic, and BTLA and then completed NASA-TLX along with a usability questionnaire. None of the participants were experts in teleoperation, but all received proper training in the system's use before the experiments. All participants gave informed consent and were briefed on the nature of the tasks they would be performing.

The evaluation task required coordinated dual-arm transport of a large object to a designated platform (see Fig. 6), covering grasp, lift/transport, and placement phases. Each participant experienced three teleoperation modes in counterbalanced order: SPB, Dyadic, and BTLA. After each trial, participants completed NASA-TLX [20] and a short usability questionnaire. Randomization minimized order effects across teleoperation modes. As depicted in Fig. 6, the task required robust dual-arm coordination to convey the object to a specified goal while preserving its pose and preventing contact with the environment. Panels (a)-(d) trace the progression from the initial stance to a secure grasp, and panels (e)-(f) show the subsequent transfer to the designated platform. The experimental conditions are summarized in Table I.

TABLE I
EXPERIMENTAL CONDITIONS DEFINITION.

Condition	Control Method	Operator(s)	Assistance
SPB	Dual haptic devices	Single	None
Dyadic	Single haptic each	Two	Human partner
BTLA	Haptic + Voice	Single	LLM assistant

D. Assessment

We measured completion time, success rate, and coverage of the target area. A trial was deemed successful only if (1) both arms grasped the object correctly, (2) transport occurred without collisions or drops, and (3) placement achieved at least 70% target coverage. Multiple trials per teleoperation mode were recorded to assess consistency. System usability and operator experience were assessed through two complementary questionnaires. Usability was rated on a multi-item Likert instrument, covering naturalness, satisfaction, perceived intelligence, and overall usability (see Fig. 7). The second employed the NASA-TLX to quantify workload on six subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration, as shown in Fig. 8.

To evaluate the LLM robustness, we conducted additional experiments analyzing system performance under adverse conditions. Voice commands were tested with varying background noise levels (45 - 75 dB), simulating industrial environments. Recognition accuracy decreased from 94% in quiet conditions to 71% at 75 dB noise level, with average latency increasing from 320 to 580 ms. Common misinterpretation patterns included confusion between "*Follow*" and "*Fetch*" commands (12% error rate) and numerical distance parameters (e.g., "two" vs "to"). We implemented a confirmation mechanism for ambiguous commands, reducing critical errors by 67%. Under noisy conditions, the system defaults to requesting verbal confirmation before executing potentially hazardous operations.

V. RESULTS AND DISCUSSION

A. Performance Metrics

Fig. 9 demonstrates that BTLA achieved the highest mean coverage (0.861) and success probability (0.627) beyond Dyadic and SPB, which indicates that the assistant-driven

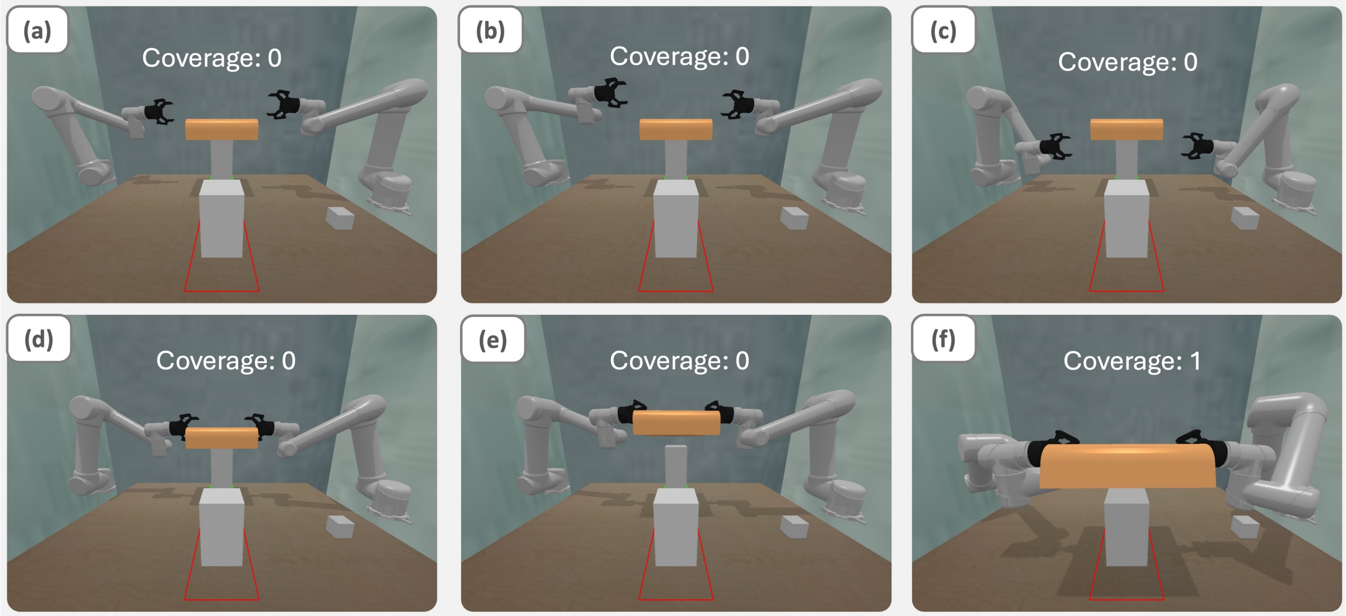

Fig. 6. Execution sequence of BTLA for an object-transfer task: (a) initial configuration; (b) left arm moves independently with follow mode disabled; (c) right arm, controlled by BTLA, performs symmetrical-following behavior; (d) both arms synchronize to reach the pick-up pose; (e) object grasp; (f) cooperative transport with BTLA to the designated target location.

Statement		Score									
How do you feel about the performance of the intelligent agent?	Unnatural	1	2	3	4	5	6	7	8	9	Natural
	Unpleasant	1	2	3	4	5	6	7	8	9	Pleasant
	Strange	1	2	3	4	5	6	7	8	9	Typical
	Dislikeable	1	2	3	4	5	6	7	8	9	Likeable

Fig. 7. Post-trial questionnaire results: six-dimension usability scores recorded on a 9-point Likert scale.

workflow more reliably completes the bimanual transport and placement task. The keyword-based system achieved a 52% task success rate, significantly lower than BTLA. Users required 3.2 times more command attempts due to rigid syntax requirements. Failures predominantly occurred when context or pronoun resolution mattered (e.g., “move it a bit closer”) and could not be captured by fixed grammar rules.

The Kruskal–Wallis test confirmed that differences in coverage ($p = 0.003$) and success rate ($p = 0.004$) across the three modes were statistically significant. In contrast, the improvement in completion time did not reach significance ($p = 0.117$), despite BTLA’s lower median times. A correlation analysis demonstrated in Fig. 10 further showed a strong positive association between coverage and success ($r = 0.708$): trials that explored a greater fraction of the task area were more likely to finish successfully. Thus, natural-language-mediated, variable autonomy primarily boosts effectiveness (coverage, success) and reliability, with time benefits emerging, though not uniformly across participants or scenarios.

B. Subjective Assessment

Statistical power analysis was conducted using G*Power 3.1. For our within-subjects design with $n = 10$ participants and three conditions, we achieved 80% power to detect large effect sizes (Cohen’s $d = 0.84$) at $\alpha = 0.05$. Post-hoc analysis revealed our observed effect sizes for coverage ($d = 1.24$) and success rate ($d = 1.08$) exceeded the MDE, validating our sample size adequacy. For medium effects ($d = 0.5$), 16 participants would be required; however, our large observed effects justify the current sample.

Across all NASA-TLX dimensions, including mental demand (MD), physical demand (PD), temporal demand (TD), performance (P), effort (E), and frustration (F), the BTLA condition yielded the best scores: reduced demands, effort, and frustration, together with higher perceived performance than both Dyadic and SPB (see Fig. 8). By contrast, SPB was the most taxing, showing elevated demands, effort, and frustration, and the lowest perceived performance. Dyadic occupied a middle ground, with intermediate workload and performance ratings between BTLA and SPB.

The Kruskal-Wallis tests indicated statistically significant differences among the three modes for each dimension (MD: $\chi^2 = 17.974$, $p < 0.001$; PD: $\chi^2 = 14.701$, $p = 0.001$; TD: $\chi^2 = 12.276$, $p = 0.0002$; P: $\chi^2 = 15.723$, $p < 0.001$; E: $\chi^2 = 14.228$, $p = 0.0001$; F: $\chi^2 = 11.018$, $p < 0.001$). Over 40% of participants attributed residual difficulty primarily to the restricted 2D view, citing depth-perception limits and occasional occlusion.

Furthermore, to validate BLTA’s real-world applicability, we deployed it on a physical dual-arm UR3e robotic system equipped with 6-DoF force/torque sensors at each end-effector. The system utilizes two Force Dimension Omega

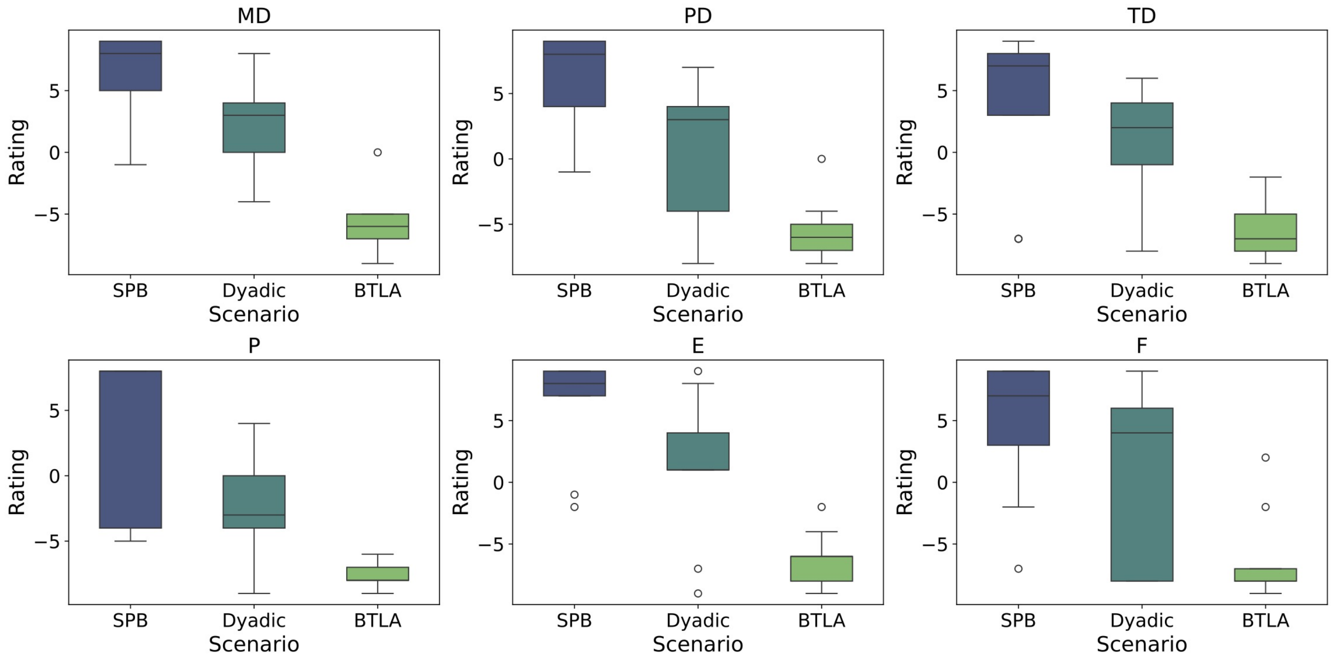

Fig. 8. NASA-TLX results summarized with boxplots for SPB, Dyadic, and BTLA (all participants). Rated aspects: mental demand (MD), physical demand (PD), temporal demand (TD), performance (P), effort (E), and frustration (F). All metrics follow statistical significance, i.e., $p < 0.05$.

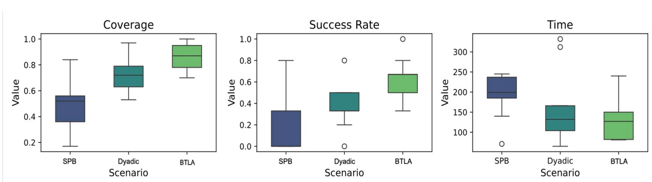

Fig. 9. Boxplots depicting performance for SPB, Dyadic, and BTLA across participants. (a) coverage ($p < 0.05$), (b) success rate ($p < 0.05$), (c) completion time ($p = 0.117$).

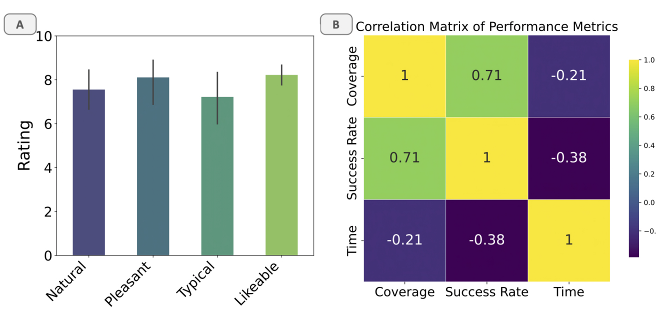

Fig. 10. (**A**) Likert Scale Ratings. (**B**) Correlation matrix of performance metrics.

7 haptic devices with bilateral force feedback. The robots were mounted in a 45-degree upside-down configuration, and transformation matrices were applied to correct the kinematic mapping between operator input and robot motion. We implemented velocity control with exponential smoothing to enhance stability while preserving responsiveness. Force measurements from the robot end-effectors were appropriately transformed and scaled before being rendered as haptic feedback to the operator. The system demonstrated robust performance across 10 trials with an 80% success rate in object manipulation tasks. BTLA maintained superior performance in physical trials despite additional challenges, including network latency (12 ± 3 ms), kinematic singularities (encountered in 20% of trials). In soft bottle handling tasks (Fig. 11, a-d), the system maintained consistent force control within ± 2.5 N of target values, preventing deformation while allowing safe manipulation. For box transportation (Fig. 11, e-h), the system successfully coordinated dual-arm movements with a 90% success rate (compared to dyadic 40% and SPB 60% in 10 trials) and completed transfers in an average of 12.3 s. Voice commands were processed with 94% recognition accuracy and an average response latency of 320 ms, enabling seamless collaboration.

VI. Conclusion

BTLA combines natural-language interaction with adjustable autonomy for single-operator dual-arm teleoperation. Across simulated and physical evaluations it consistently raised task success and coverage while reducing operator workload, handling ambiguous phrasing, and supporting smooth skill transitions. Physical trials on a dual-UR3e platform confirmed robustness to network delays and kinematic singularities.

Several promising directions for future work emerge from this study. Firstly, expanding the range of autonomous behaviors and developing more sophisticated real-time autonomy adaptation could further enhance the system's flexibility. Secondly, investigating how operators and the assistant system adapt to each other during extended use could provide insights for improving human-robot collaboration. Thirdly,

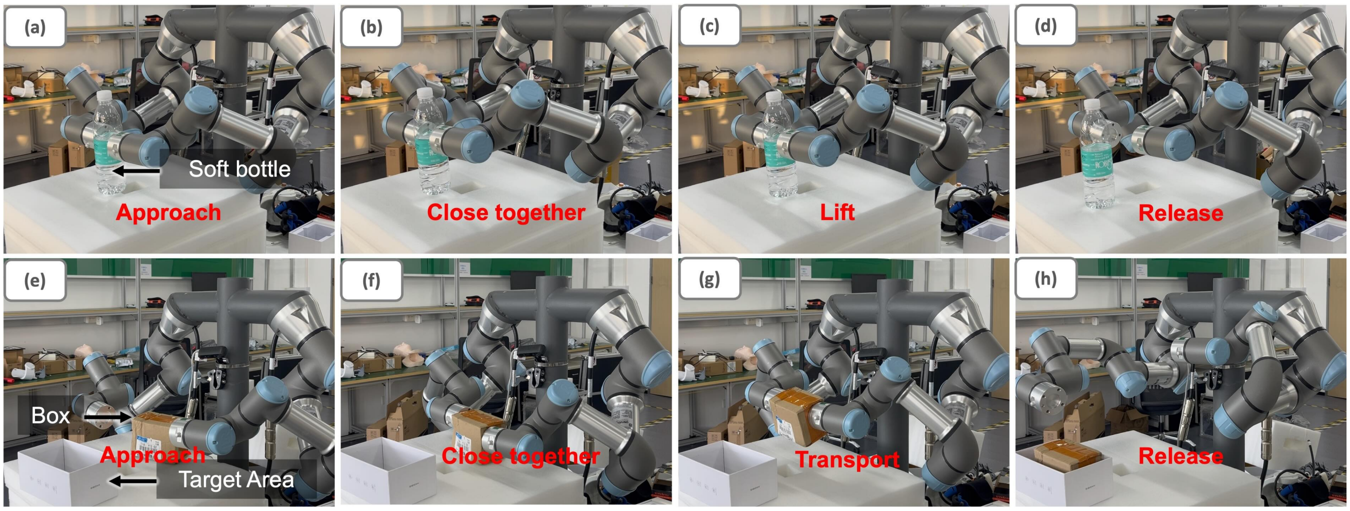

Fig. 11. Sequential demonstration of BTLA-assisted manipulation tasks in real-world scenarios. Top row (a-d): Collaborative manipulation of a soft bottle through approach, grasping, lifting, and release phases. Bottom row (e-h): Box transportation task showing approach, coordination, transport, and placement in the target area.

long-term adaptation presents significant opportunities for BTLA enhancement. Future implementations could incorporate reinforcement learning from operator feedback, enabling skill refinement based on usage patterns.

Acknowledgment

We thank all participants for their engagement and support throughout the study.

References

[1] Z. Wang, H.-K. Lam, B. Xiao, Z. Chen, B. Liang, and T. Zhang, "Event-triggered prescribed-time fuzzy control for space teleoperation systems subject to multiple constraints and uncertainties," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 9, pp. 2785–2797, 2021.

[2] Y. Sun, X. Chen, Z. Wang, H. Qin, and R. Jing, "Adaptive interval type-2 fuzzy control for multi-legged underwater robot with input saturation and full-state constraints," *International Journal of Systems Science*, vol. 54, no. 15, pp. 2859–2874, 2023.

[3] J. R. Boehm, N. P. Fey, and A. M. Fey, "Online recognition of bimanual coordination provides important context for movement data in bimanual teleoperated robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6248–6255.

[4] Y. Guo, D. Freer, F. Deligianni, and G.-Z. Yang, "Eye-tracking for performance evaluation and workload estimation in space telerobotic training," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 1, pp. 1–11, 2022.

[5] W. Bai, Z. Wang, Q. Cao, H. Yokoi, M. G. Fujie, E. M. Yeatman, and G.-Z. Yang, "Anthropomorphic dual-arm coordinated control for a single-port surgical robot based on dual-step optimization," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 1, pp. 72–84, 2022.

[6] Z. Wang, H. Fei, Y. Huang, Q. Rouxel, B. Xiao, Z. Li, and E. Burdet, "Learning to assist bimanual teleoperation using interval type-2 polynomial fuzzy inference," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 2, pp. 416–425, 2024.

[7] S. Shao, T. Wang, Y. Wang, Y. Su, C. Song, and C. Yao, "Research of HRV as a measure of mental workload in human and dual-arm robot interaction," *Electronics*, vol. 9, no. 12, p. 2174, 2020.

[8] E. Noohi, M. Žefran, and J. L. Patton, "A model for human–human collaborative object manipulation and its application to human–robot interaction," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 880–896, 2016.

[9] Y. Che, G. M. Haro, and A. M. Okamura, "Two is not always better than one: Effects of teleoperation and haptic coupling," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 1290–1295.

[10] G. Li, F. Caponetto, V. Katsageorgiou, N. G. Tsagarakis, and I. Sagakoglou, "A telerobotic system enabling online switching among various architectures and controllers," *Robotics and Autonomous Systems*, vol. 166, p. 104402, 2023.

[11] A. García, J. E. Solanes, A. Muñoz, L. Gracia, and J. Tornero, "Augmented reality-based interface for bimanual robot teleoperation," *Applied Sciences*, vol. 12, no. 9, p. 4379, 2022.

[12] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, p. eaaw0955, 2019.

[13] C. Zhou, L. Zhao, H. Wang, L. Chen, and Y. Zheng, "A bilateral dual-arm teleoperation robot system with a unified control architecture," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 495–502.

[14] D. Sun, Q. Liao, and A. Loutfi, "Single master bimanual teleoperation system with efficient regulation," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1022–1037, 2020.

[15] J. Pan, J. Eden, D. Oetomo, and W. Johal, "Effects of shared control on cognitive load and trust in teleoperated trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5863–5870, 2024.

[16] E. Turco, C. Castellani, V. Bo, C. Pacchierotti, D. Prattichizzo, and T. L. Baldi, "Reducing cognitive load in teleoperating swarms of robots through a data-driven shared control approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 4731–4738.

[17] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 523–11 530.

[18] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, P. Li, Y. Zheng, Z. Xia, K. Zhan, X. Lang, Y. Chen *et al.*, "Planagent: A multi-modal large language agent for closed-loop vehicle motion planning," *arXiv preprint arXiv:2406.01587*, 2024.

[19] I. Akinola, J. Xu, S. Song, and P. K. Allen, "Dynamic grasping with reachability and motion awareness," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9422–9429.

[20] S. G. Hart and L. E. Staveland, *Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research*. Elsevier, 1988, vol. 52.