Lessons Learned From the euROBIN Manipulation Skill Versatility Challenge At IROS 2024

Xudong Han, *Student Member, IEEE*, Haoran Sun, *Student Member, IEEE*, Ning Guo, *Member, IEEE*, Sheng Ge, *Student Member, IEEE*, Jia Pan, *Member, IEEE*, Fang Wan, *Member, IEEE*, and Chaoyang Song, *Senior Member, IEEE*

***Abstract*—Robotic automation in high-mix, low-volume settings often relies on structured workcells where fixtures, workpieces, and procedures are well defined, but tasks must be reconfigured quickly across variants. This article presents the Design and Learning Research Group's solution to the euROBIN manipulation skill versatility challenge at IROS 2024, in which a standardized electronic task board is operated through a fixed sequence of subtasks. We deliberately adopt a simple, industry-style implementation: each subtask is taught by recording an open-loop tool center point and gripper trajectory in the task-board frame, a wrist-mounted depth camera is used once per run to localize the board and a few key features, and a web dashboard exposes the resulting motion primitives as reusable code blocks that can be reordered and reparameterized. With this pipeline, our system completes the benchmark in 28.2 s in the lab (37.2 s on-site), compared to a human baseline of 16.3 s and an average of 83.5 s for previous teams, closing a large part of the performance gap without advanced learning or feedback control. We further demonstrate limited skill transfer by retargeting the same subtask library to a smoke-detector battery replacement scenario required by the competition. Finally, we release our implementation as open source and discuss how competition design and benchmark structure influence the balance between simple engineered solutions and richer sensing and learning.**

***Index Terms*—Manipulation skills, robotic versatility.**

I. Introduction

ROBOTIC automation in modern manufacturing and testing often takes place in structured workcells, where fixtures and procedures are well defined, but product variants and workflows change over time. In this context, "versatility" is less about open-ended cognition and more about how quickly a system can be reconfigured across related tasks without extensive reprogramming or hardware changes [1], [2], [3], [4]. We adopt this concrete notion of versatility in this article and focus on how a standardized manipulation task can be decomposed into reusable subtasks (STs) that are easy to reorder and retarget.

The euROBIN manipulation skill versatility challenge (MSVC), held annually since 2021 (known as Robothon from 2021 to 2023), embodies this type of practice-oriented versatility [5], [6], [7]. It uses a standardized electronic task board and a fixed sequence of STs that mimic an industrial testing workflow, including object localization, insertion, door operation, circuit probing, and cable management. Under the official protocol, human operators complete the task board in 16.3 s, whereas prior robotic entries report average completion times around 83.5 s, with substantial variability across teams.

At IROS 2024 in Abu Dhabi, the Design and Learning Research Group (DLRG) participated for the first time, as shown in Fig. 1. We intentionally adopted a simple, industry-style approach: each ST is taught by direct demonstration as an open-loop tool center point (TCP) and gripper trajectory in the task-board frame; a wrist-mounted depth camera and a YOLO-based detector are used once per run to estimate the board pose and a few key features; and a web-based dashboard exposes the resulting motion primitives as modular code blocks with adjustable parameters. With this implementation, our system completes the MSVC benchmark in 28.2 s in the lab and 37.2 s on-site, substantially closing the gap to the human baseline without relying on advanced learning algorithms or sophisticated feedback control.

Beyond the official task board, we reuse the same ST library, with minor retargeting, to demonstrate a smoke-detector battery replacement scenario required by the organizers. In this work, we therefore use the term "skill" to denote parameterized, reusable motion primitives rather than learned policies, and we characterize "skill transfer" as the reuse and retargeting of this library within a family of structurally related fixtures. Other forms of versatility that are crucial in less structured domains, such as manipulation of unknown objects, operation in cluttered household environments, or following high-level instructions from nonexpert users [5], [8], [9], [10], [11], [12], [13], [14], [15], are outside the scope of this article and will likely require richer perception, planning, and learning capabilities.

Received 15 December 2024; revised 22 February 2025 and 20 November 2025; accepted 19 December 2025. Date of publication 24 December 2025; date of current version 15 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62206119 and Grant 62473189 and in part by the Shenzhen Long-Term Support for Higher Education at SUSTech under Grant 20231115141649002. This article was recommended for publication by Associate Editor I. Lutkebohle upon evaluation of the reviewers' comments. *(Corresponding authors: Fang Wan; Chaoyang Song.)*

Xudong Han, Ning Guo, Sheng Ge, and Fang Wan are with the Southern University of Science and Technology, Shenzhen 518055, China (e-mail: 12231112@mail.sustech.edu.cn; 11930729@mail.sustech.edu.cn; 12132253@mail.sustech.edu.cn; wanfang@ieee.org).

Haoran Sun is with the Southern University of Science and Technology, Shenzhen 518055, China, and also with the University of Hong Kong, Hong Kong SAR, China (e-mail: 12150050@mail.sustech.edu.cn).

Jia Pan is with the University of Hong Kong, Hong Kong SAR, China (e-mail: jpan@cs.hku.hk).

Chaoyang Song is with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates (e-mail: songcy@ieee.org).

The webpage of this article is available online at https://msvc-dlrg.github.io.

This article has supplementary downloadable material available at https://doi.org/10.1109/RAP.2025.3648290, provided by the authors.

Digital Object Identifier 10.1109/RAP.2025.3648290

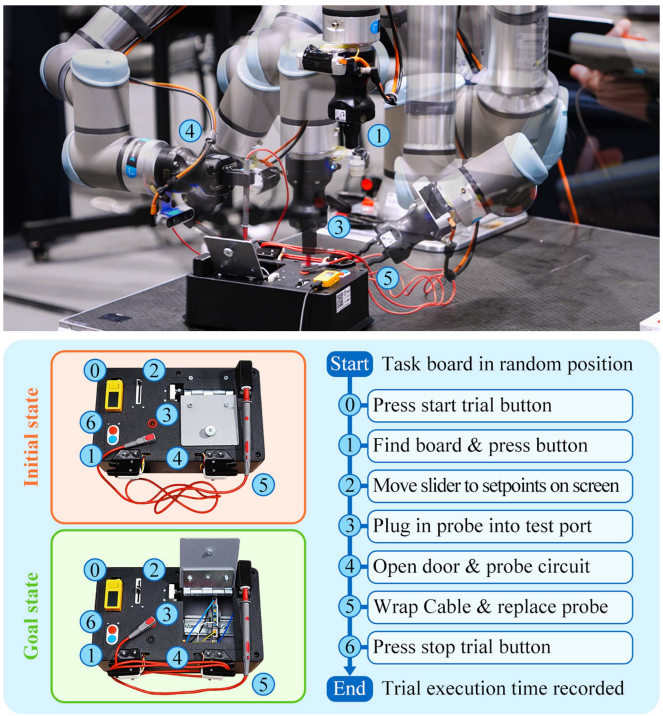

Fig. 1. Design and Learning Research Group's implementation of the 2024 euROBIN MSVC, including the on-site hardware deployment in Abu Dhabi, task board in initial and goal states, and the predefined task procedures for the competition.

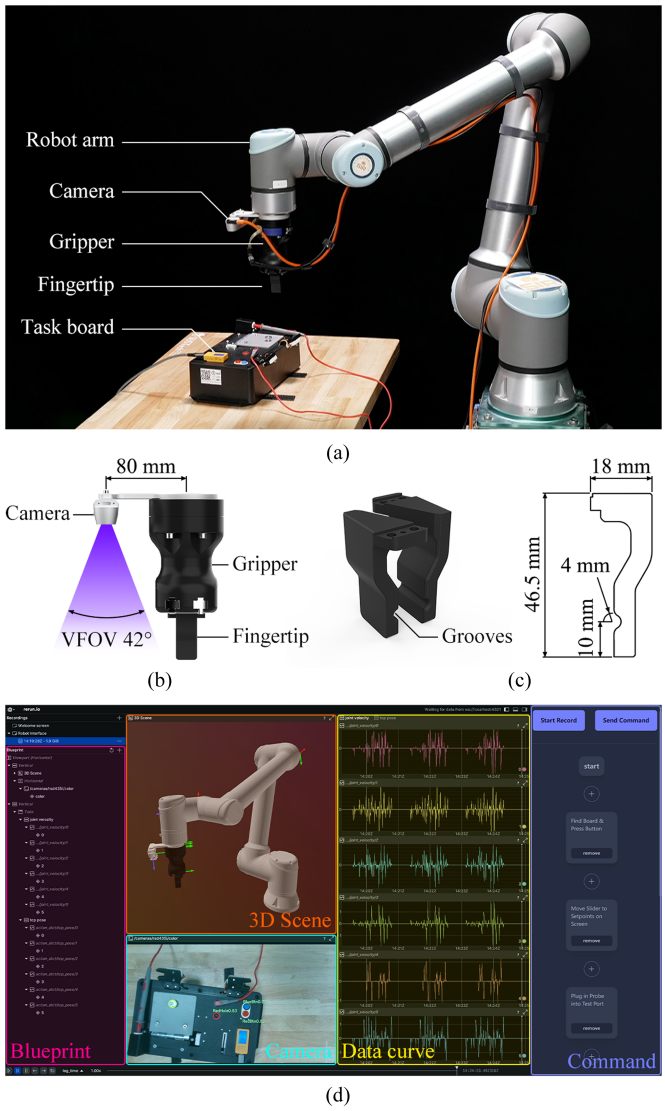

Fig. 2. *Robot platform design*. (a) Hardware composition. (b) Detailed view of the wrist camera. (c) Customized fingertip design. (d) Customized user interface for displaying robot data and rapid task execution.

II. Methods

A. Robot System Design

Fig. 2(a) shows the robot system developed by DLRG for the competition, including a robot arm (UR10e, Universal Robots), a camera (D435, Intel), a gripper (Hand-E, Robotiq), and a pair of 3-D-printed fingertips with minor customization for cable handling, all mounted on the open-sourced DeepClaw station [16]. The camera was mounted on a bracket between the gripper and the robot arm flange, with a horizontal center-to-flange distance of 80 mm. This is primarily determined by the vertical field of view of the camera being 42°, ensuring the camera observes the area below and in front of the gripper at an appropriate distance without obstruction [as shown in Fig. 2(b)]. Considering combining the cables in ST 5, we redesigned the original fingertip of the gripper by adding a groove of 4 mm diameter [as shown in Fig. 2(c)], fabricated using nylon (PA12, HP) through multijet fusion.

We also designed a web-based user interface, as shown in Fig. 2(d). On the left side of the interface was a *blueprint* panel outlining the robot arm's joints and links, the camera, and the gripper with fingertips. The middle includes a *3-D scene* of the robotic arm showing its current state in Cartesian space, a *camera* view of the live video stream taken from the wrist overlayed with key feature recognition status, and a stack of *data curve* windows showing the robot's angles and velocities in the joint space. The right is a *command* panel with code blocks for each ST skill (e.g., grasping, dragging, inserting, and pressing) and a drag-and-drop interface to rapidly restructure and refine the overall task flow with adjustable parameters. We also added an extra record button to start and stop recording robot data for convenient execution and redeployment.

B. Locating the Task Board

Accurately locating the task board is crucial for successful task execution. Since the Velcro strips firmly fix the task board at the back, localization only needs to be estimated *once* before the task begins. Fig. 3(a) illustrates the coordinate systems of the robot arm $\mathcal{F}_\mathrm{r}$, camera $\mathcal{F}_\mathrm{c}$, gripper $\mathcal{F}_\mathrm{g}$, and task board $\mathcal{F}_\mathrm{b}$. The transformation from $\mathcal{F}_\mathrm{r}$ to $\mathcal{F}_\mathrm{g}$, ${}^{\mathrm{r}}_{\mathrm{g}}\mathbf{T}$, is determined by the robot arm's joint settings, while ${}^{\mathrm{r}}_{\mathrm{c}}\mathbf{T}$ from $\mathcal{F}_\mathrm{r}$ to $\mathcal{F}_\mathrm{c}$ is obtained via hand-eye calibration. For $\mathcal{F}_\mathrm{b}$, the origin is at the blue button, the y-axis extends to the red button, and the x-axis is perpendicular to the y-axis [see Fig. 3(b)]. The transformation between $\mathcal{F}_\mathrm{c}$ and $\mathcal{F}_\mathrm{b}$ is defined as

$$\begin{bmatrix} {}^{\mathrm{c}}\mathbf{x}_i \\ 1 \end{bmatrix} = {}^{\mathrm{c}}_{\mathrm{b}}\mathbf{T} \begin{bmatrix} {}^{\mathrm{b}}\mathbf{x}_i \\ 1 \end{bmatrix} \tag{1}$$

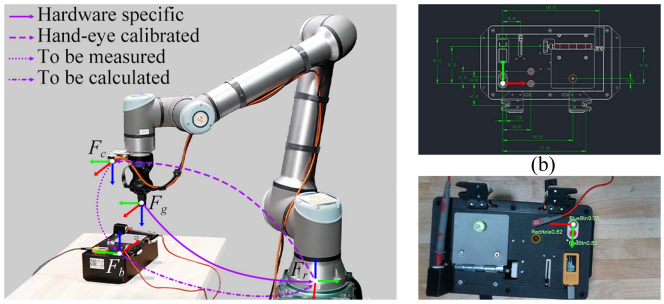

Fig. 3. *Locating the task board.* (a) Configuration of the robot system. (b) Premeasured template of the task board. (c) Estimated locations by detecting the red and blue buttons and the red test port.

where $\{{}^{\mathrm{c}}\mathbf{x}_i, {}^{\mathrm{b}}\mathbf{x}_i | i = 1, 2, 3\}$ are coordinate pairs of features (red button, blue button, and red test port) in $\mathcal{F}_{\mathrm{c}}$ and $\mathcal{F}_{\mathrm{b}}$. The objective is to find the least-squares estimate

$$\min \frac{1}{3} \sum_{i=1}^{3} \| {}^{\mathrm{c}}_{\mathrm{b}}\mathbf{R} \cdot {}^{\mathrm{b}}\mathbf{x}_i + {}^{\mathrm{c}}_{\mathrm{b}}\mathbf{t} - {}^{\mathrm{c}}\mathbf{x}_i \|. \tag{2}$$

With the task board template measured, ${}^{\mathrm{b}}\mathbf{x}_i$ is known [see Fig. 3(b)], whereas ${}^{\mathrm{c}}\mathbf{x}_i$ is estimated from camera images [see Fig. 3(c)] using YOLO [17], trained on 1000 labeled images achieving over 75% classification confidence score. Utilizing the perspective projection model $\mathbf{\Pi}$ and camera matrix $\mathbf{A}$, pixel coordinates $\mathbf{p}_i = (u_i, v_i)$ from YOLO are converted to $\mathcal{F}_{\mathrm{c}}$ points

$$\begin{bmatrix} {}^{\mathrm{c}}\mathbf{x}_i \\ 1 \end{bmatrix} = \mathbf{\Pi}^{-1}\mathbf{A}^{-1} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}. \tag{3}$$

Therefore, ${}^{\mathrm{c}}_{\mathrm{b}}\mathbf{T}$ is obtained using (2) and (3) with ${}^{\mathrm{b}}\mathbf{x}_i$ and $\mathbf{p}_i$. ${}^{\mathrm{r}}_{\mathrm{b}}\mathbf{T} = {}^{\mathrm{r}}_{\mathrm{c}}\mathbf{T}\,{}^{\mathrm{c}}_{\mathrm{b}}\mathbf{T}$ is calculated to translate the TCP trajectories under $\mathcal{F}_{\mathrm{b}}$ for robot execution.

C. Teaching Robot for Tasks

Robot teaching enables rapid and stable implementation of structured tasks and is widely used in industrial practice. In this work, we intentionally refrain from any advanced learning or closed-loop control and instead implement all manipulation STs as open-loop trajectories taught by direct demonstration. Using the teach pendant, the operator manually guides the robot through the desired TCP motion and gripper actions for each ST (e.g., pressing, inserting, dragging, and wrapping), and the resulting trajectories are recorded in the robot base frame $\mathcal{F}_{\mathrm{r}}$.

Because the task board can be mounted at different positions, the transformation ${}^{\mathrm{b}}_{\mathrm{r}}\mathbf{T}$ between the robot base frame $\mathcal{F}_{\mathrm{r}}$ and the board frame $\mathcal{F}_{\mathrm{b}}$ changes from run to run. For execution, we therefore express all control commands relative to $\mathcal{F}_{\mathrm{b}}$ and convert them to $\mathcal{F}_{\mathrm{r}}$ using the board's estimated pose. Specifically, a trajectory point recorded in $\mathcal{F}_{\mathrm{r}}$, denoted ${}^{\mathrm{r}}\mathbf{x}_i$, is transformed to the board frame as

$$\begin{bmatrix} {}^{\mathrm{b}}\mathbf{x}_i \\ 1 \end{bmatrix} = {}^{\mathrm{b}}_{\mathrm{r}}\mathbf{T} \begin{bmatrix} {}^{\mathrm{r}}\mathbf{x}_i \\ 1 \end{bmatrix} \tag{4}$$

where $\{{}^{\mathrm{b}}\mathbf{x}_i \mid i = 1, 2, \ldots, n\}$ and $\{{}^{\mathrm{r}}\mathbf{x}_i \mid i = 1, 2, \ldots, n\}$ denote the same discretized TCP trajectory expressed in the board and robot frames, respectively. During execution, we recompute ${}^{\mathrm{b}}_{\mathrm{r}}\mathbf{T}$ from the current board localization (see Section II-B), and replay the recorded trajectories in an open-loop fashion at the desired speed.

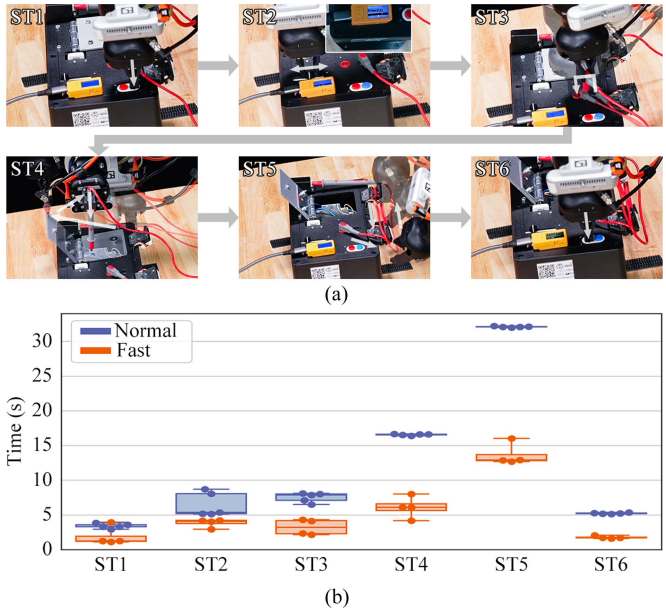

Fig. 4. *Trial process and time of each ST.* (a) Snapshots of the six STs. (b) ST time during trials at normal and fast speeds.

Each ST is encapsulated as a code block with precondition and postcondition (e.g., "probe grasped" or "button pressed"), but in our competition system, these conditions are not verified online; the sequence is predefined to maximize execution speed. Our custom dashboard [see Fig. 2(d)] exposes the taught STs as drag-and-drop blocks with adjustable parameters, such as approach offsets, dwell times, and velocity scaling. This design reflects common industrial practices, in which operators use proprietary graphical interfaces rather than low-level code. As no open-source platform provided the required functionality, we developed and released our own implementation for this competition online.[1]

In the remainder of this article, we use the term "skill" to refer to these parameterized, reusable motion primitives rather than to learned policies. Skill transfer in our experiments corresponds to reusing and retargeting this library of open-loop STs to a structurally similar fixture, as demonstrated in Section III-B. Extending this framework to learned or feedback-based skills would require additional sensing and computation, but could build on the same abstraction and interface.

III. Results

A. Robust and Fast Completion of Tasks

Following the competition protocol, we performed five consecutive trials at 1 m/s, randomly repositioning the task board to test robustness (see video 3 of the Supplementary Material). Fig. 4(a) shows the six STs: ST1 and ST6 pressed and released the button; ST2 positioned the slider using YOLO and color thresholding; ST3 moved the probe from black to red test port; ST4 opened the door using the probe and inserted it into the testing slot; and ST5 wrapped the cable.

We then increased speed to 2.5 m/s for three successful high-speed trials and demonstrated on-site success at 2 m/s.

[1]GitHub: https://github.com/ancorasir/DesignLearnRG_euROBIN

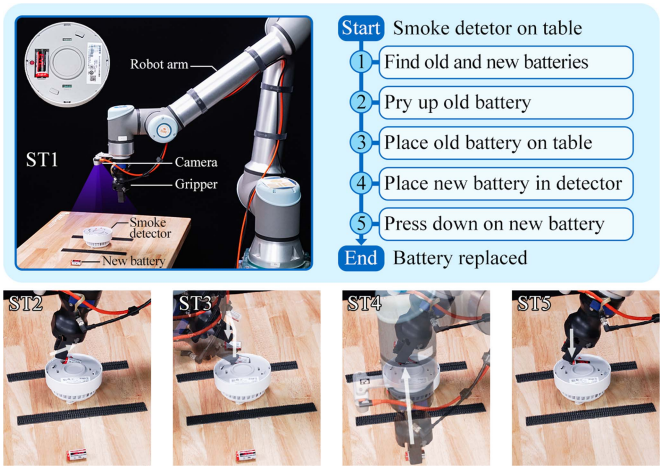

Fig. 5. Skill transfer to replacing the battery in a smoke detector.

Fig. 4(b) compares *normal* (1 m/s) versus *fast* ($\geq$ 2 m/s) speeds: at normal speed, execution times were at least 1.7 times slower, with standard deviations mostly under 0.7 s except ST2 (1.75 s) due to pointer position randomness. Fast speeds increased standard deviations, with effects influenced by trajectory density, top-speed reachability, and recognition feedback delays. These results confirm the system's reliability and efficiency across varying conditions.

B. Skill Transfer to Battery Replacement

The implemented task-board skills can be reused in a structurally related scenario: replacing the battery of a smoke detector, which is part of the competition requirements. Rather than learning new policies, we decompose this scenario into five STs that directly reuse previously taught skills with adjusted TCP trajectories and gripper configurations (see Fig. 5). ST1 uses YOLO to detect the battery and estimate its pose; ST2 pries the old battery out of its holder using a motion analogous to opening the task-board door; ST3 and ST4 pick and place batteries using motions similar to probe insertion and removal, respectively; and ST5 presses the new battery into its final position using a button-pressing motion.

All STs in this battery replacement demo are therefore still executed in open loop, using the same parameterized code blocks created for the task board and retargeted to a new set of key poses. This constitutes a limited but practically relevant form of skill transfer: once a library of robust STs exists for a standardized fixture, a new but geometrically similar use case can be implemented primarily by redefining a small number of waypoints rather than rewriting the entire control program. The process, shown in the video 4 of the Supplementary Material, was performed multiple times during preparation, although only one representative trial was retained per the competition rules. A more systematic evaluation with greater hardware and environmental variability is left for future work.

IV. Discussion & Conclusion

This article documents an implementation practice for the euROBIN MSVC at IROS 2024. Our system completes the

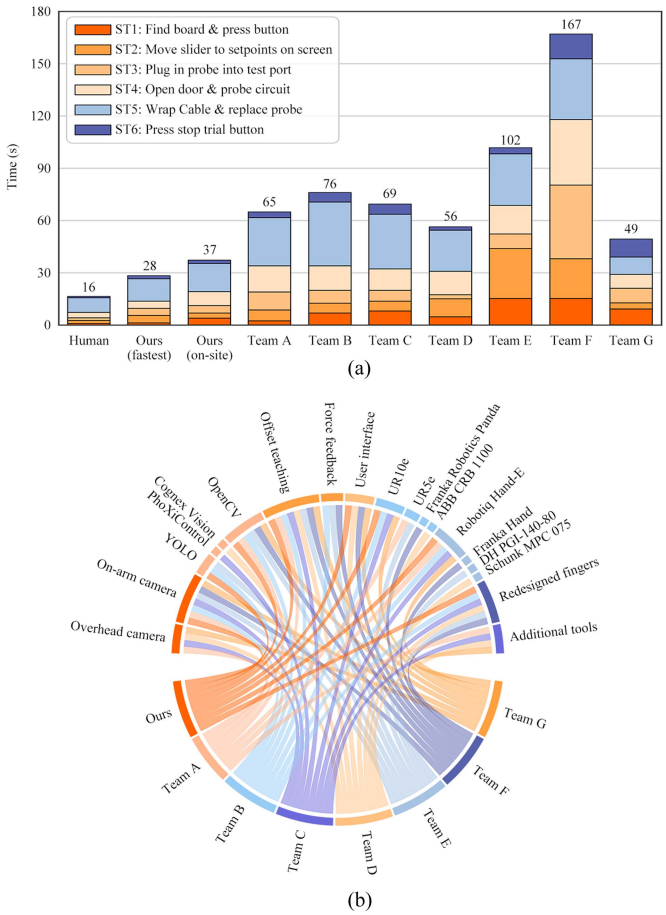

Fig. 6. *Comparison on execution time and team solutions*. (a) Stacked bar plot of trial execution time showing the performance of humans, our system, and seven other teams. (b) Chord diagram showing the teams' strategy and platform component selections, including sensing modalities, end-effector choices, and degrees of feedback control.

standardized electronic task board in 28.2 s in the lab and 37.2 s on-site, compared to a human baseline of 16.3 s and an average of 83.5 s reported for previous teams. All data on other teams were collected from public videos, web pages, and task board logs. Rather than a controlled scientific study, the work is intended as a transparent case study of how a simple, industry-style solution can close much of the performance gap on a structured benchmark.

As summarized in Fig. 6(a), our system achieved shorter execution times in all STs than the teams reported in the official logs, with ST1 (initial localization and button press), ST4 (door opening and probe insertion), and ST5 (cable handling) particularly close to human performance. Public information suggests that several teams employed additional sensing modalities (for example, multiple RGB-D cameras or force sensing) and more conservative motion strategies, such as slower joint velocities, approach-and-correct phases, or explicit contact detection. These design choices can improve robustness and generality but tend to increase cycle time. Our main qualitative finding is that, for a highly structured and repeatable benchmark, such as MSVC, a carefully engineered open-loop implementation can be both fast and sufficiently robust, provided that the task board is rigidly mounted and its geometry is well characterized.

The discrepancy between our lab and on-site performance further illustrates the importance of mechanical design. In the lab, all hardware was mounted on the DeepClaw station [16], which is heavy and rigid enough to support high-speed motions without noticeable vibration. At the competition venue, the hardware had to be mounted on a much lighter table. Running the UR10e at the same speed caused visible oscillations of the table and task board, making the motion unsafe and forcing us to reduce velocity and acceleration. Vision-related effects, such as lighting variation and minor calibration changes, played at most a secondary role. This experience aligns with common engineering practice: structural rigidity can be as critical as algorithmic sophistication when deploying high-speed robotic cells.

The competition's design also shapes the kinds of solutions favored. While preparing for the competition, despite the availability of more advanced tools, such as soft robotic fingers with omnidirectional adaptation and state-of-the-art tactile sensing [18], [19], we still chose a more structured and industrial solution that emphasizes task completion efficiency over intelligence integration or exploration. As tasks become more constrained and scoring emphasizes execution time, there is a natural tendency for teams to converge toward efficient but relatively inflexible methods, as previously observed in competitions, such as the amazon picking challenge [20], [21]. To encourage advances in perception, learning, and adaptive manipulation, future competitions and testbeds could introduce more diversity in objects, dynamic changes in the environment, or evolving ST sequences. Our implementation can then serve as a strong baseline for these extended scenarios, against which richer sensing and learning methods can be evaluated.

Finally, this study clarifies one specific dimension of robotic versatility. In the literature, the term has been used to describe diverse phenomena [22], [23], [24], [25], including robots whose morphology supports many behaviors [26], [27], task-level generalization and policy reuse [28], and economical reconfiguration of production systems [1], [2]. Our work does not claim cognitive adaptability or human-level versatility. Instead, it contributes an open, reproducible example of rapid reconfiguration within a family of closely related tasks, using reusable, frame-consistent motion primitives, and a simple interface. Future work will explore how to extend by integrating tactile sensing and feedback-based skills, learning policies that automatically adapt or sequence STs from demonstration data, and formalizing metrics that distinguish different flavors of versatility in both industrial and competition settings.

REFERENCES

[1] N. Dhanaraj et al., "A human robot collaboration framework for assembly tasks in high mix manufacturing applications," in *Proc. Int. Manuf. Sci. Eng. Conf.*, vol. 87240, 2023, Art. no. V002T07A011.

[2] T. Tahmina, M. Garcia, Z. Geng, and B. Bidanda, "A survey of smart manufacturing for high-mix low-volume production in defense and aerospace industries," in *Proc. Int. Conf. Flexible Automat. Intell. Manuf.*, 2022, pp. 237–245.

[3] G. Bravo-Palacios, A. Del Prete, and P. M. Wensing, "One robot for many tasks: Versatile co-design through stochastic programming," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1680–1687, Apr. 2020.

[4] Q. Ze et al., "Magnetic shape memory polymers with integrated multifunctional shape manipulation," *Adv. Mater.*, vol. 32, no. 4, 2020, Art. no. 1906657.

[5] F. Amigoni et al., "Competitions for benchmarking: Task and functionality scoring complete performance assessment," *IEEE Robot. Automat. Mag.*, vol. 22, no. 3, pp. 53–61, Sep. 2015.

[6] B. Calli, A. Dollar, M. A. Roa, S. Srinivasa, and Y. Sun, "Guest editorial: Introduction to the special issue on benchmarking protocols for robotic manipulation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8678–8680, Oct. 2021.

[7] P. So, A. Sarabakha, F. Wu, U. Culha, F. J. Abu-Dakka, and S. Haddadin, "Digital robot judge: Building a task-centric performance database of real-world manipulation with electronic task boards," *IEEE Robot. Automat. Mag.*, vol. 31, no. 4, pp. 32–44, Dec. 2024.

[8] J. Cui and J. Trinkle, "Toward next-generation learned robot manipulation," *Sci. Robot.*, vol. 6, no. 54, 2021, Art. no. eabd9461.

[9] F. Wang, J. R. G. Olvera, and G. Cheng, "Optimal order pick-and-place of objects in cluttered scene by a mobile manipulator," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6402–6409, Oct. 2021.

[10] R. Madhavan, R. Lakaemper, and T. Kalmár-Nagy, "Benchmarking and standardization of intelligent robotic systems," in *Proc. IEEE Int. Conf. Adv. Robot.* (ICAR), 2009, pp. 1–7.

[11] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Automat. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[12] T. Hajar, A. Ghassane, and Z. Abdelhamid, "Workforce assignment problem considering versatility in a collaborative robot system," in *Proc. Int. Conf. Integr. Des. Prod.*, 2022, pp. 551–564.

[13] M. Zhao, K. Okada, and M. Inaba, "Versatile articulated aerial robot DRAGON: Aerial manipulation and grasping by vectorable thrust control," *Int. J. Robot. Res.*, vol. 42, no. 4/5, pp. 214–248, 2023.

[14] M. Dyck et al., "Toward safe and collaborative robotic ultrasound tissue scanning in neurosurgery," *IEEE Trans. Med. Robot. Bionics*, vol. 6, no. 1, pp. 64–67, Feb. 2024.

[15] K. Karacan, R. J. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin, "The inherent representation of tactile manipulation using unified force-impedance control," in *Proc. IEEE Conf. Decis. Control*, 2023, pp. 4745–4752.

[16] F. Wan, H. Wang, X. Liu, L. Yang, and C. Song, "DeepClaw: A robotic hardware benchmarking platform for learning object manipulation," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron. (AIM)*, 2020, pp. 2011–2018.

[17] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[18] N. Guo et al., "Proprioceptive state estimation for amphibious tactile sensing," *IEEE Trans. Robot.*, vol. 40, pp. 4662–4676, 2024.

[19] X. Liu, X. Han, W. Hong, F. Wan, and C. Song, "Proprioceptive learning with soft polyhedral networks," *Int. J. Robot. Res.*, vol. 43, no. 12, pp. 1916–1935, 2024.

[20] N. Correll et al., "Analysis and observations from the first Amazon picking challenge," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 1, pp. 172–188, Jan. 2018.

[21] D. Morrison et al., "Cartman: The low-cost Cartesian manipulator that won the Amazon robotics challenge," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7757–7764.

[22] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annu. Rev. Control*, vol. 49, pp. 113–127, 2020.

[23] S. Hong, Y. Um, J. Park, and H.-W. Park, "Agile and versatile climbing on ferromagnetic surfaces with a quadrupedal robot," *Sci. Robot.*, vol. 7, no. 73, 2022, Art. no. eadd1017.

[24] C. Mastalli et al., "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2536–2542.

[25] P. J. Chen and D. R. Liu, "Prime editing for precise and highly versatile genome manipulation," *Nature Rev. Genet.*, vol. 24, no. 3, pp. 161–177, 2023.

[26] M. Zhao et al., "Versatile multilinked aerial robot with tilted propellers: Design, modeling, control, and state estimation for autonomous flight and manipulation," *J. Field Robot.*, vol. 38, no. 7, pp. 933–966, 2021.

[27] X. Shu et al., "A versatile humanoid robot platform for dexterous manipulation and human–robot collaboration," *CAAI Trans. Intell. Technol.*, vol. 9, no. 2, pp. 526–540, 2024.

[28] R. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, and C. Schmid, "Learning to combine primitive skills: A step towards versatile robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4637–4643.