RAMP: A Benchmark for Evaluating Robotic Assembly Manipulation and Planning

Jack Collins, *Member, IEEE*, Mark Robson, Jun Yamada, *Graduate Student Member, IEEE*, Mohan Sridharan, *Member, IEEE*, Karol Janik, and Ingmar Posner, *Member, IEEE*

***Abstract*—We introduce RAMP, an open-source robotics benchmark inspired by real-world industrial assembly tasks. RAMP consists of beams that a robot must assemble into specified goal configurations using pegs as fasteners. As such, it assesses planning and execution capabilities, and poses challenges in perception, reasoning, manipulation, diagnostics, fault recovery, and goal parsing. RAMP has been designed to be accessible and extensible. Parts are either 3D printed or otherwise constructed from materials that are readily obtainable. The design of parts and detailed instructions are publicly available. In order to broaden community engagement, RAMP incorporates fixtures such as April Tags which enable researchers to focus on individual sub-tasks of the assembly challenge if desired. We provide a full digital twin as well as rudimentary baselines to enable rapid progress. Our vision is for RAMP to form the substrate for a community-driven endeavour that evolves as capability matures.**

***Index Terms*—Performance evaluation and benchmarking, assembly, manipulation planning, task and motion planning.**

I. Introduction

ROBOTS are predicted to raise manufacturing productivity globally by $0.8 - 1.4\%$ annually [1]. Of particular interest in this domain are offsite construction tasks, where parts are largely pre-assembled offsite and only installed in-situ. Offsite construction is attractive because it supports a substantial reduction in total build time through parallel production of modular assemblies alongside site works [2]. In the United States alone, offsite construction is predicted to increase from 33.64% to 54.9% in the next decade with automation levels set to increase by 7% over the same period [3]. However, while beams for the internal structure of a building constructed offsite are often produced through light-steel-roll-forming before they are cut and punched to size – a process that is readily automated – the assembly of beams into frames remains a manual process. What complicates matters is that there is a high likelihood of every assembly being different (see Fig. 2). Automating the assembly process itself requires robust perception, versatile skill primitives, long-horizon task and motion planning, robust action execution, and fault recovery. It touches on the full gamut of robotics research. Delivering this capability requires a focused and concerted effort.

One mechanism for fostering such focus are benchmarks. Benchmarks have had significant impact in catalysing progress across a number of different domains such as object detection [4], visual odometry [5] and reinforcement learning [6].

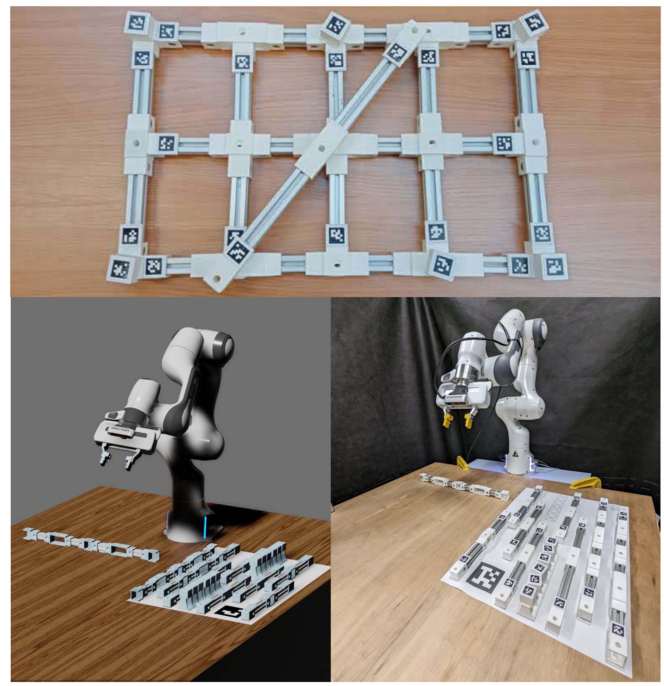

Fig. 1. *Top:* The benchmark beams and pegs with fiducial markers attached. *Bottom-Left:* Simulation environment with fixed beam on the left, additional beams and pegs laying on a layout template, Panda robot with optimised fingertips and wrist camera, and birds-eye camera looking down from above. *Bottom-Right:* Physical setup of the benchmark with the same setup as the simulation environment.

Manuscript received 16 May 2023; accepted 17 October 2023. Date of publication 6 November 2023; date of current version 16 November 2023. This letter was recommended for publication by Associate Editor N. Kong and Editor C.-B. Yan upon evaluation of the reviewers' comments. This work was supported in part by UKRI/EPSRC Programme under Grant EP/V000748/1, in part by UK EPSRC under Grant EP/S032487/1, and in part by the US Office of Naval Research Award under Grant N00014-23-1-2535. *(Jack Collins, Mark Robson, and Jun Yamada contributed equally to this work.) (Corresponding author: Jack Collins.)*

Jack Collins, Jun Yamada, and Ingmar Posner are with the Applied AI Lab, Oxford Robotics Institute, University of Oxford, OX1 3PJ Oxford, U.K. (e-mail: jcollins@robots.ox.ac.uk; jyamada@robots.ox.ac.uk; ingmar@robots.ox.ac.uk).

Mark Robson is with the University of Birmingham, B152TT Birmingham, U.K. (e-mail: mxr880@student.bham.ac.uk).

Mohan Sridharan is with the Intelligent Robotics Lab, School of Computer Science, University of Birmingham, B152TT Birmingham, U.K. (e-mail: m.sridharan@bham.ac.uk).

Karol Janik is with The Manufacturing Technology Centre, CV7 9JU Coventry, U.K. (e-mail: karol.janik@the-mtc.org).

Digital Object Identifier 10.1109/LRA.2023.3330611

2377-3766 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

Fig. 2. Light steel-rolled beams automatically cut and punched to size, and manually assembled and fastened. Such assemblies take a lot of time and effort as each beam and assembly can be unique within a prefabricated building.

However, the design of benchmarks for robotics is notoriously challenging. For one, robots are often heterogeneous across research labs, comprising many, often bespoke, interacting components and subsystems. This renders a typical like-for-like comparison of methods difficult. In addition, there is considerable disparity in the resources available across research labs working on robotics problems. Finally, real-world problems of commercial interest, such as robotic assembly, require a full systems solution, incorporating perception, planning, and control. In contrast, researchers may want to engage only in particular sub-tasks aligned with their specific research interests to drive progress. All of these factors present significant barriers to entry.

This letter presents the Robotic Assembly Manipulation and Planning (RAMP) benchmark. Situated in the space of offsite construction tasks, it is designed to mitigate the common pitfalls in robotics benchmarks mentioned above. Specifically, it is designed to be challenge-driven, accessible, and open-ended.

Challenge-Driven: RAMP closely mimics offsite construction challenges. It presents three categories of assemblies (*easy*, *medium*, and *hard*) along with an evaluation protocol that reflects the need for repeatable and fast assembly of parts as well as metrics that promote speed and completeness.

Accessible: RAMP consists of 3D printed parts and extruded aluminium profiles (see Fig. 1, top), materials that are widely available. The part designs and detailed instructions are open-source and publicly available at the benchmark website[1]. In addition, RAMP distributes with an accurate simulation environment in Nvidia Isaac [7] that mirrors the real-world setup. To enable researchers to focus on individual sub-tasks if desired, RAMP components have been instrumented to simplify scene perception. Similarly, a baseline method has been implemented both in simulation and the real-world for researchers either to build upon or to use as substrate in conjunction with their own sub-task solutions.

Open-Ended: While RAMP provides a number of goal configurations to work with, the parts can be used to create many other configurations. Entirely new parts can be readily created to broaden the spectrum of reachable configurations. While the parts have been designed such that goal configurations are achievable using a single arm, the RAMP benchmark readily lends itself to more complex setups such as dual-arm manipulation. In the future, we envision extensions also requiring tactile sensing and deformable object manipulation. Our vision is for RAMP to grow into a community-driven effort which evolves as capability matures.

In the following sections we review previously proposed robotic manipulation benchmarks (Section II), explore the proposed RAMP benchmark (Section III), review the implementation and performance of our baseline approach (Section IV) and present directions the benchmark could expand into (Section V).

II. Literature Review

Benchmarks serve an important function in stimulating research progress in key areas by enabling measurement of progress and direct comparison between methods [8]. Although many robotic manipulation benchmarks have been proposed [8], [9], [10], [11], [12], relatively few focus on robotic assembly [13]. Among the benchmarks for robotic assembly [8], [14], [15], [16], [17], only a subset assess long-horizon task success and real-world performance [18], [19]. The other robotic assembly benchmarks can be viewed as assessing performance with short-horizon tasks or in simulated environments alone.

Short-horizon robotic assembly tasks, e.g., peg-in-hole problems, have received a lot of attention with current methods achieving high success rates [20]. However, long-horizon robotic assembly tasks present many additional challenges. Traditionally, methods for such long-horizon tasks use prior knowledge and observations to search for a sequence of manually-encoded primitive actions ("skills") that result in the desired goal [21]. Recent work has also developed data-driven methods to jointly learn actions and plans [22]. Many frameworks for automated assembly have been investigated and assessed on a range of application spaces including Lego [21], IKEA furniture [16], [23], [24], timber framing [25], industrial parts [26], and other profiles [27]. As these methods for automated assembly have often been assessed on unique assembly tasks that are hard to access, it is difficult to directly compare them. Our benchmark provides a common set of easily accessible parts that existing frameworks can be evaluated upon, enabling users to directly compare methods.

Among benchmarks for multi-step real-world robotic assembly, the NIST assembly boards assess completion of a set of industry-related assembly tasks, each involving one of four defined boards with fixed goal configurations. A points-based scoring system weights components added (or subtracted), with points for parts correctly seated, inserted, clipped, threaded, and routed on a board [18]. The tasks cover a range of manipulation challenges including peg-in-hole problems, gear meshing, nut threading, flexible parts handling and more. The simpler boards do not require reasoning about assembly sequencing as parts may be added or subtracted in any order, making the manipulation skills the focus of the challenge. Also, as the assembly boards are composed of many distinct parts, it can be difficult

[1] [Online]. Available: https://sites.google.com/oxfordrobotics.institute/ramp/create-your-own

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

to create a new board from scratch, limiting the extent to which the benchmark is accessible and open-ended.

Another multi-step real-world robotic assembly benchmark is the bimanual semi-deformable object benchmark [19]. This benchmark is split into two separate tasks with performance assessed based on success rate and time. The first task is inspired by watchmaking; it involves assembling 3D printable parts and requires multiple interactions to insert a single part into another. The second task is rubber-band manipulation, which also requires multiple fine-motor control interactions. However, there is no requirement for high-level task planning or reasoning as the tasks are sequential with a single set goal. Additionally, there is a hardware constraint mandating the use of a bimanual platform which limits accessibility of the benchmark to research groups with such a system.

Many competitions run at robotics conferences in past years have assessed the participants' ability to solve long-horizon assembly tasks. This includes the FetchIt! mobile manipulation challenge [28], the Industrial Assembly Challenge run multiple times at the World Robot Summit [29] and the Robotic Grasping and Manipulation Competition that uses the NIST assembly boards [30]. Such competitions provide a motivating, competitive setting for research groups to assess performance on a common set of tasks, oftentimes using the same hardware. However, although competitions form a good basis for a benchmark assessing long-horizon robot assembly, the parts are not easily accessible to non-competitors and therefore there remains a need for an open-source, accessible alternative.

Common themes across the reviewed robotic benchmarks and competitions include a lack of widely accessible benchmarking components, and systems that assess the ability of a method to achieve a fixed goal at the expense of generalisation to new, similar goal configurations. Our benchmark, RAMP, seeks to overcome these limitations of existing benchmarks and competitions while building on their desirable attributes, to make three key contributions. First, RAMP promotes accessibility by developing parts that are easy to create. Second, it is designed to be open-ended by providing parts that are reconfigurable and expandable such that the same parts can be used to assemble many unique designs, i.e., it can be used to assess the ability of a solution to generalise to new goal configurations. Finally, RAMP is designed explicitly to assess long-horizon, multi-step reasoning on physical systems, a challenging problem in robotics research.

III. Benchmark

RAMP is designed to jointly assess planning and execution for robotic assembly, with the core principles of RAMP to provide a challenge-driven, accessible and open-ended benchmark. Currently, the benchmark is intended to assess single-arm manipulation although we hope to see it grow with the community to encompass other domains. The benchmark consists of a set of predefined beams made from an extensible set of base parts, a predefined set of assemblies for comparable assessment, metrics for assessing performance, an evaluation protocol to promote reproducibility, and an accurate simulation environment for prototyping and testing. Each of these components is described further below.

A. Base Parts

The benchmark is composed of nine base beams and up to 15 pegs—see Fig. 1 (top). The beams are constructed from 20×20 mm extruded-profile aluminium with 3D-printed joints. To construct the base beams, 33 3D joints need to be printed with 24 lengths of aluminium extrusion cut to size. To assist with state estimation and beam identification, set joints have space for a 25×25 mm April Tag [31]. The joints' part files and an exact bill of parts detailing the number and type of joints to print, the lengths of aluminium profile, and an assembly guide are available on the RAMP website.

In addition to the base beams, other parts are available that can be 3D printed including fingertips designed for picking up the beams, clamps to help fix beams to a table, and holders that position pegs at a convenient pick orientation. To assist with localisation of parts and reproducible research, an A2 template is available for laying out parts, with an outline for all beams and pegs, and a large fiducial marker for state estimation. We anticipate the number of beams and assemblies to evolve to include more challenging designs over time.

B. Tasks

RAMP's goal configurations are grouped into three classes: easy, medium, and hard. Fig. 3 shows the goal configurations for each class of assemblies, with more sophisticated skills being required to complete the medium and hard classes. For example, the easy class of problems only contains $3-4$ peg insertions with the most complicated action being a capping action to complete a rectangle. For the medium class of problems, $4-8$ pegs need to be inserted, with additional skills necessary to insert a beam at an angle, slide beams through other beams, and to re-grasp beams during such actions. For the hard class of problems, there are free-form designs that require additional planning considerations and beams that need to be angled suitably before sliding them through other beams. In general, the hard class of assemblies requires a longer sequence of actions which makes them difficult to complete successfully.

In addition to the image of the goal configuration, the configurations of the beams in the goal state are described using Extensible Markup Language (XML). The XML format is easy to use with existing planners, and this process can be automated; other modalities (e.g., images, natural language) are possible and encouraged.

When taken together, these three classes of assemblies pose fundamental open problems in perception, state estimation, long-horizon planning, control and execution of primitive skills, fault recovery, and learning. These goal configurations should act as a holdout set with similar designs also achievable using the same methods.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

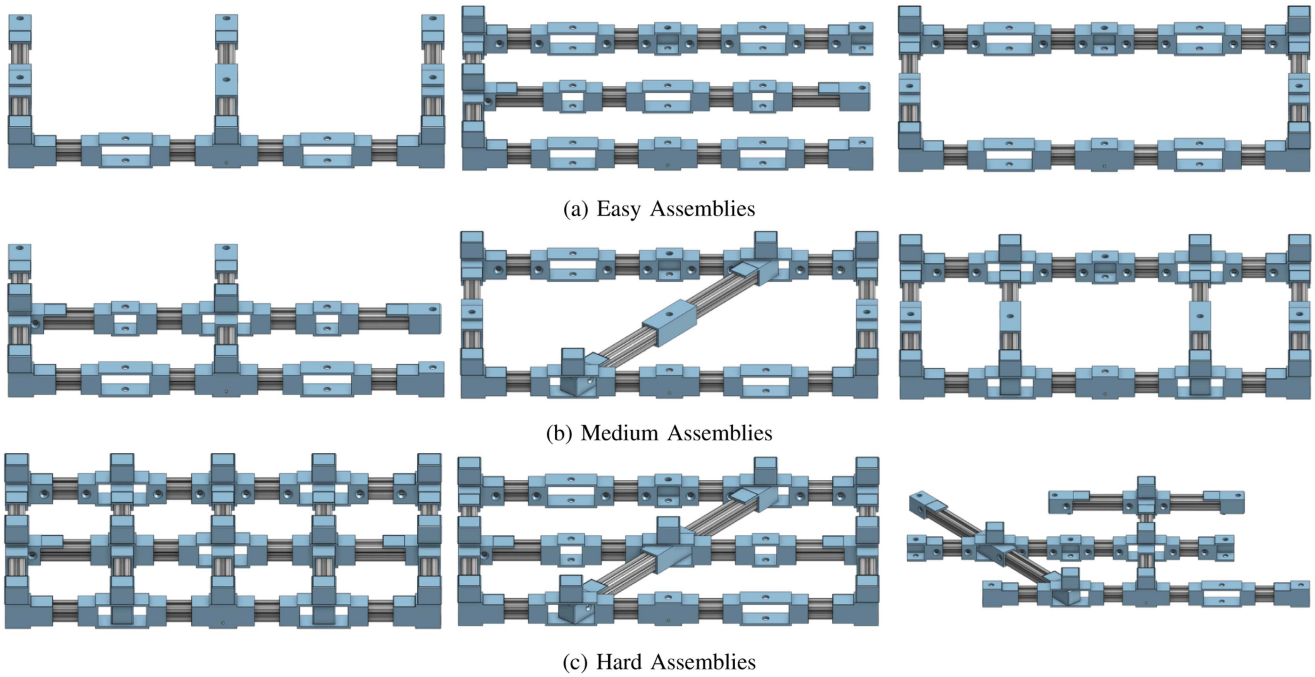

Fig. 3. Three classes of assemblies for assessing the capabilities of any proposed solution to the benchmark.

C. Performance Measures

The performance of methods being tested on the the RAMP benchmark is assessed using measures that have been chosen carefully to reflect the needs of industry. The two priorities are that the assemblies: (i) are completed accurately; and (ii) in the shortest amount of time. An insertion is accurate if a peg is inserted, such that no stem is visible, into the correct hole pinning the required two joints of the beams. If a peg is not inserted into the correct hole, is not flush with the surface of the joint, or is not pinning the correct two joints together then the task completion percentage does not increase. The results obtained from RAMP thus include the time of each successful peg insertion with the time also including the time taken to generate the executed plan. These results are presented as plots similar to those found in Fig. 5, and include the average across five repeats and the most successful attempt.

D. Evaluation Protocol

For reproducibility, we strongly suggest users adhere to our evaluation protocol. Specifically, an entire class of assemblies must be completed using the same code, with only the goal changing between assemblies. To assess robustness and repeatability of the methods being evaluated, each assembly in a class needs to be performed five times consecutively without altering the code; the only change between individual runs, as stated before, is the goal to be achieved.

To promote accessibility, RAMP is designed to be used with any robot that has a manipulator arm and sensors for state estimation. The manipulator can also use any available gripper ranging from parallel grippers to anthropomorphic hands. Diverse set of sensor setups are encouraged for state estimation as this promotes the exploration of different solution methods for the underlying perception challenges.

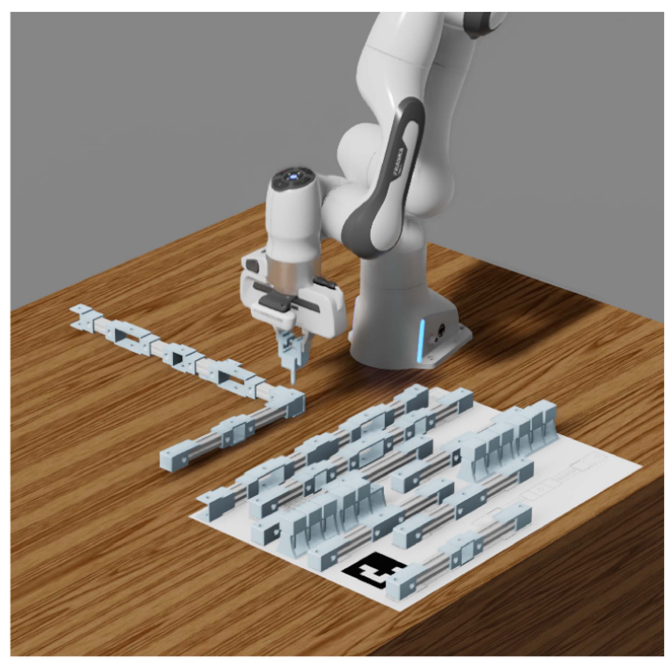

Fig. 4. Open-source RAMP simulation environment using Nvidia Isaac. Assembly in simulation uses SDF contacts.

To support the use of a single manipulator, the first beam of the assembly is fixed rigidly to the table, with all assemblies in the benchmark using it as the starting point. The fixed beam must be located within the robot's operational space. All other parts for any target assembly must start on the A2 template in

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

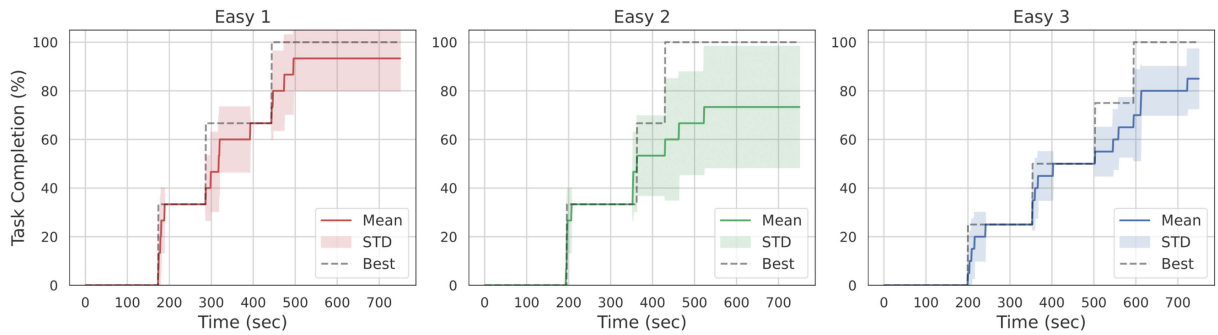

Fig. 5. Results of the baseline method across three assemblies in the easy class of the benchmark, with the percentage of completion of the desired assembly expressed as a function of the time (planning and execution, in seconds). The best and average of the repeats are plotted along with the standard deviation. Standard deviation increases with execution time since the more complex assemblies typically require more actions.

their allocated spot. The A2 template must not lie below the completed assembly but no other constraints are imposed on its starting location; Fig. 1 (bottom right) shows an example setup. From the start state, the assembly must be constructed using the available parts. More than the necessary number of pegs may be used; if a peg is dropped during assembly, another peg may be used to achieve the goal.

E. Simulation

RAMP includes a complete simulation environment created to mirror the real-world setup (see Fig. 4). The environment is built using the Nvidia Isaac simulator [7] which comes with an API, real-time physics simulation capability, and realistic renderings. Faithful collision detection based on Signed Distance Fields (SDF) [32] is used to compute fine-resolution contacts between the gripper (of the manipulator) and the beams, and for reliably simulating the insertion of joints and pegs.

The simulation environment unlocks access to a broad range of learning-based methods, and enables development and assessment of solutions without needing access to a real-world setup. The simulation provides observations including end-effector pose and force-torque data; joint position, velocity and torque; peg and beam state information; and image observations for wrist and birds-eye cameras. We have also created high-level interfaces to basic control approaches including Cartesian position and impedance control.

We anticipate multiple extensions of the RAMP simulation environment. This includes increasing the realism of the environment through the addition of observation noise and the integration of April Tags for marker-based state estimation. Additional control methods, e.g., force-based and closed-loop policies, are another opportunity to improve the environment. Finally, to expand the environment to learning-based methods, the inclusion of procedurally generated beams and assemblies as well as the ability to parallelise the environment are important directions for future work.

IV. Baseline Implementation

RAMP includes a baseline method that aims to lower the barrier to entry for new adopters and can serve as a foundation for new methods. The baseline pursues a classical approach to solve the easy class of assemblies with the integration of a task planner, conditioned on a goal assembly specified in XML, and hand-engineered skills to execute the derived plan. The baseline's components and performance are discussed below.

A. Task Planner

The baseline task planner is a simplification of an existing refinement-based architecture (REBA) that reasons with any given domain's transition diagrams at two resolutions, with the fine-resolution description defined formally as a *refinement* of the coarse-resolution description [33]. In REBA, for any given goal, non-monotonic logical reasoning at the coarse-resolution with commonsense domain knowledge computes a plan of abstract actions. Each abstract transition is implemented as a sequence of fine-resolution actions by automatically *zooming* to and reasoning (probabilistically) with the relevant part of the fine-resolution description. The fine-resolution outcomes are then mapped to abstract ones that are used for subsequent reasoning. REBA has enabled reliable, efficient, and transparent decisions on robots and simulated agents in complex domains as it supports inference, planning, and diagnostics in the presence of noisy sensing and actuation.

In our baseline, the coarse-resolution system description $\mathcal{D}_c$ comprises statements in action language $\mathcal{AL}_d$ [34] encoding the domain signature (Σ_c) and axioms governing domain dynamics. Σ_c includes a hierarchy of *sorts* such as $robot$, $place$, $thing$, $object$, $beam$, and $step$ (for temporal reasoning); actions such as $move(robot, place)$, $pick_up(robot, beam)$, and $assemble(robot, beam)$; statics, i.e., domain attributes such as $next_to(place, place)$ that cannot be changed; and fluents, i.e., domain attributes that can be changed, e.g., $loc(thing, place)$ and $in_hand(robot, thing)$. Axioms in $\mathcal{D}_c$ include *causal laws*, *state constraints*, and *executability conditions* exemplified by the following examples:

$$putdown(R, O) \textbf{ causes } \neg\, in_hand(R, O)$$

$$loc(O, Pl) \textbf{ if } loc(R, Pl),\ in_hand(R, O)$$

$$\textbf{impossible } pick_up(R, O) \textbf{ if } in_hand(R, O)$$

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

where variables *R*, *O*, and *Pl* represent the robot, an object, and a place respectively.

To compute a plan of abstract actions to achieve a given goal, our baseline constructs and solves a program $\Pi(\mathcal{D}_c, \mathcal{H}_c)$ in Answer Set Programming (ASP) [35] that encodes $\mathcal{D}_c$, the initial conditions, a history $\mathcal{H}_c$ of prior observations and action executions, and heuristic methods and helper axioms that direct the search for the plan. ASP encodes concepts such as *default negation* and *epistemic disjuction*, supports non-monotonic logical reasoning, and reduces reasoning to computing *answer set(s)* of the corresponding program. Each answer set of $\Pi(\mathcal{D}_c, \mathcal{H}_c)$ includes a sequence of abstract actions to reach the goal. To compute answer sets, we use the SPARC system [36] and the *clingo* SAT solver [37].

To map abstract transitions to actions that can be executed by the robot, our baseline supports a fine-resolution system description $\mathcal{D}_f$, a refinement of $\mathcal{D}_c$, which can be viewed as looking at $\mathcal{D}_c$ through a magnifying lens to discover structures that were previously abstracted away. The sorted signature of $\mathcal{D}_f$ includes sorts for these new structures and those in $\mathcal{D}_c$, and actions, statics, and fluents described in terms of these sorts, e.g., $assemble(robot, object_part)$ and $in_hand(robot, object_part)$. The axioms are refined in a similar manner, and additional *bridge axioms* link the attributes in $\mathcal{D}_c$ and $\mathcal{D}_f$. For each abstract transition, $\Pi_f(\mathcal{D}_f, \mathcal{H}_f)$ is solved to extract answer sets with sequences of fine-resolution actions. The planning code for our baseline is available through the RAMP website[2].

The ordered list of fine-resolution actions output by the planner are to be executed by the robot in an open loop. Specifically, the following assembly actions are used:

- *move(robot, place):* to move the end effector to a location.
- *pick_up(robot, part):* to grasp a part with the gripper.
- *put_down(robot, part):* to release a part from the gripper.
- *assemble_square(robot, joint):* to insert beams where a specified joint fits into an existing joint in the assembly.
- *assemble_cap(robot, joint):* to insert a beam such that one or more joints in the assembly fit into the beam.
- *fasten(robot, joint, joint, peg):* to insert a peg into two beams at their connecting joints.
- *push(robot, beam):* to correct the position of beams already in the assembly.

The implementation of these actions is discussed below.

B. Action Execution

Any given plan is executed using hand-designed skills that are intentionally designed to act as the simplest approach. Each fine-resolution action has a corresponding skill that is executed without re-planning. Sensors (e.g., wrist-mounted and bird's eye cameras) are used to observe the April tags [31] on the beams and pegs, estimating their location in world coordinates. Since the offsets of these tags to each joint and link of the beams are known, joint and link poses can be computed in the global reference frame.

[2][Online]. Available: https://sites.google.com/oxfordrobotics.institute/ramp

Our baseline uses the *MoveIt* [38] motion planner for executing simple actions such as *move*, *pick_up*, and *put_down*, and uses Cartesian-space impedance control and force feedback for executing contact-rich actions.For example, to complete the peg-in-hole skill for the assembly action *fasten*, force feedback is essential to detect any contact between the peg and the beam surface. The robot moves downwards until a specific z-axis force threshold is reached, indicating contact between the peg and the beam surface. A spiral search algorithm [39] is then employed to locate the hole position. After the peg is aligned with the hole, the robot repeatedly moves forward and backward to prevent jamming during insertion. If unsuccessful in detecting the hole, the robot moves upwards to re-estimate the hole position using the nearest fiducial marker observed by the wrist camera and re-attempts the insertion.

The *assemble_square* assembly action inserts the grasped beam into another beam already in the assembly. This skill requires the robot to move linearly until the grasped beam makes contact with the other beam, detected using force feedback. Then, the joint of the other beam in the assembly is localised by moving the grasped beam along the face of contact in a 1D implementation of the spiral search algorithm. After localising the joint in the assembly, the robot aligns the two beams using force feedback.

The assembly action of *assemble_cap* similarly uses Cartesian impedance control to move the grasped beam towards the desired pose until a reactionary force, exerted against the robot, exceeds a threshold value. After capping, alignment of the beams is achieved by the assembly action *push*. The *push* action corrects the position of a beam aiming to reposition an inserted beam to the nominal target position in order to ensure alignment with future beams that might be inserted into the assembly. This skill is implemented using Cartesian impedance control and executed until a force threshold is reached.

C. Results

We evaluated the baseline method using the evaluation protocol from Section III, with the results presented in Fig. 5. Fig. 6 shows snapshots of the actions and plan adopted by the baseline for an assembly, and further videos are available on the RAMP website. Fig. 5 depicts the proposed metric of task completion percentage versus time (in seconds) across the three easy assemblies. Each subplot plots the average, best and standard deviation over the five repeats. To summarise the performance of our baseline, we compute the average of the mean successes and the mean time across the three assemblies, which is $84\%/580$ sec. We observe that planning takes $180 - 200$ seconds. Given the simplicity of the baseline, we expect more sophisticated methods to achieve a higher success rate in a shorter period of time.

A significant portion of failure cases in these experiments stem from the peg insertion action. The robot struggles to accurately estimate the hole due to observation noise and is unable to correct for slanted pegs held within the gripper. Also, when the robot fails to achieve precise alignment of two beams, insertion of the peg into the designated hole is not possible. The peg insertion action is one of the more difficult actions within the

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

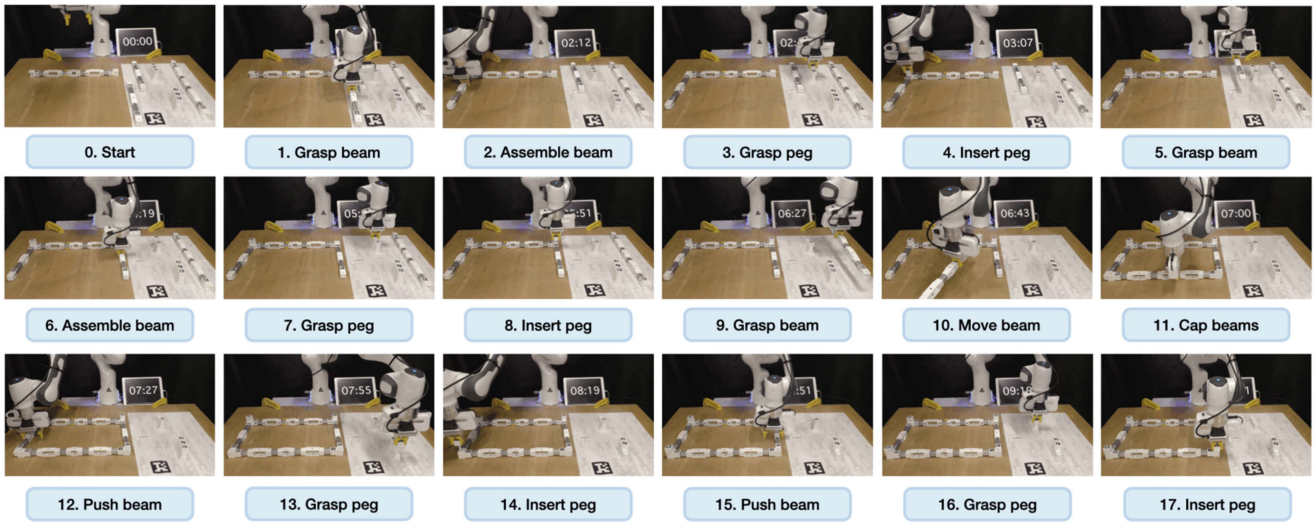

Fig. 6. Visualisation of a plan attempted by a robot using the baseline method for the third easy assembly from Fig. 3(a).

benchmark but could be made more reliable by modeling and reasoning about the uncertainty in perception and actuation, and by introducing learning-based methods like InsertionNet that provide high success rates on challenging insertion problems [20]. Fig. 5 also shows that the standard deviation increases with time, implying that assemblies that require a greater number of actions result in lower success rates. Improving the integration between perception, planning and action execution is essential to improve the observed performance on the easy class of assemblies and to consistently complete the more complex assemblies in the medium and hard classes of RAMP. This can be done by improving the baseline method or by developing other sophisticated methods. Furthermore, to attempt the medium and hard classes of RAMP, the set of assembly actions used by the planner needs to be extended, e.g., to handle rotated beams, and all fine-resolution actions will need to be enhanced to achieve a higher success rate.

V. Looking Ahead

There are many opportunities to improve upon the proposed baseline implementation to overcome the limitations and see vast improvements in the results. One such improvement could be to extend the task planner to support diagnostics in order to recover from failure cases by detecting faults and re-planning. Further to this point, tighter coupling between the task planner and the motion planner in the form of a better integration of the continuous representations necessary for robot control and the discrete representations used for higher-level task planning, would lead to improvements in the generation and successful execution of the sequences of skills for various tasks [40].

There are many avenues to increase the robustness and speed of the manipulation skills, such as by using learned motion plans [41], and combining motion planning with learned contact-rich manipulations [42], [43]. Integrating additional modalities of sensing could also see improvements in skill reliability, with one promising option being tactile sensing as touch is a crucial sensory input relied upon by humans to assemble similar objects [44].

RAMP supports easy extension to additional open challenges in robotics, some of which we discuss here. The benchmark is currently designed to be completed by a single-arm system but it could easily be extended to assess multi-agent planning and dual-arm coordination [19] by removing the rigidly attached beam and using systems with multiple manipulators. Another important extension is to assess disassembly rather than assembly as the search space for assembly can be reduced by first solving the disassembly problem [45]. The assembly problem can also be extended to 3D by designing additional parts that build upon the base set of parts, this would bring with it additional challenges in path planning and collision avoidance. In addition, extending the simulation environment as discussed in Section III-E would open additional opportunities for a separate virtual benchmark and enable learning-based assembly methods. Finally, removing the fiducial markers from the beams would extend the benchmark to include challenges in object identification and pose estimation whilst making the benchmark a more accurate representation of the real-world problem.

VI. Conclusion

RAMP is a challenge-driven, accessible, and open-ended benchmark. It is industry-insipred with the objective of catalysing the research community toward making progress on open challenges related to robot assembly. To make the benchmark easily accessible, the physical components are open-sourced and made from easy-to-source 3D printed materials and extruded aluminium profiles. These same parts can be reconfigured into any number of designs, making the range of assemblies open-ended. RAMP is distributed with a rudimentary baseline implemented both on real-world hardware and in an accurate simulation environment. Adopters of RAMP can choose to develop a full systems solution or build upon the provided baseline to drive progress in areas aligned with their interests. In the

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.

future, we plan to further develop the proposed baseline method and to expand the benchmark to support other settings such as bimanual manipulation. Our vision is for RAMP to form the foundation for a community-driven endeavour, which evolves as capability matures.

Acknowledgment

For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

References

[1] J. Manyika et al., "A future that works: AI, automation, employment, and productivity," McKinsey Global Institute Res., San Francisco, CA, USA, Tech. Rep. 60, pp. 1–135, 2017.

[2] R. E. Smith, "Off-site and modular construction explained," *Nat. Inst. Building Sci.*, 2016.

[3] R. H. Assaad, I. H. El-adaway, M. Hastak, and K. L. Needy, "Quantification of the state of practice of offsite construction and related technologies: Current trends and future prospects," *J. Construction Eng. Manage.*, vol. 148, no. 7, 2022, Art. no. 04022055.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[6] G. Brockman et al., "OpenAI gym," 2016, *arXiv:1606.01540*.

[7] NVIDIA, "Nvidia Isaac Sim." Accessed: Mar. 23, 2023. [Online]. Available: https://developer.nvidia.com/isaac-sim

[8] 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.

[9] H. Mnyusiwalla et al., "A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1389–1396, Apr. 2020.

[10] I. Garcia-Camacho et al., "Benchmarking bimanual cloth manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1111–1118, Apr. 2020.

[11] J. Collins, J. McVicar, D. Wedlock, R. Brown, D. Howard, and J. Leitner, "Benchmarking simulated robotic manipulation through a real world dataset," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 250–257, Jan. 2020.

[12] A. Suarez, V. M. Vega, M. Fernandez, G. Heredia, and A. Ollero, "Benchmarks for aerial manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2650–2657, Apr. 2020.

[13] Y. Jiang, Z. Huang, B. Yang, and W. Yang, "A review of robotic assembly strategies for the full operation procedure: Planning, execution and evaluation," *Robot. Comput.- Integr. Manuf.*, vol. 78, 2022, Art. no. 102366.

[14] T. Yu et al., "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Proc. Conf. Robot Learn.*, 2019, pp. 1094–1100.

[15] L. Fan et al., "SURREAL: Open-source reinforcement learning framework and robot manipulation benchmark," in *Proc. Conf. Robot Learn.*, 2018, pp. 767–782.

[16] Y. Lee, E. S. Hu, and J. J. Lim, "IKEA furniture assembly environment for long-horizon complex manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6343–6349.

[17] Y. Li, K. Mo, L. Shao, M. Sung, and L. Guibas, "Learning 3 D part assembly from a single image," in *Proc. Comput. Vis.: 16th Eur. Conf.*, 2020, pp. 664–682.

[18] K. Kimble et al., "Benchmarking protocols for evaluating small parts robotic assembly systems," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 883–889, Apr. 2020.

[19] K. Chatzilygeroudis et al., "Benchmark for bimanual robotic manipulation of semi-deformable objects," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2443–2450, Apr. 2020.

[20] O. Spector and D. D. Castro, "InsertionNet - A scalable solution for insertion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5509–5516, Jul. 2021.

[21] L. Nägele, A. Hoffmann, A. Schierl, and W. Reif, "Legobot: Automated planning for coordinated multi-robot assembly of lego structures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9088–9095.

[22] S. K. S. Ghasemipour, S. Kataoka, B. David, D. Freeman, S. S. Gu, and I. Mordatch, "Blocks assemble! Learning to assemble with large-scale structured reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 7435–7469.

[23] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham, "Can robots assemble an IKEA chair?," *Sci. Robot.*, vol. 3, no. 17, 2018, Art. no. eaat6385.

[24] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "IkeaBot: An autonomous multi-robot coordinated furniture assembly system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 855–862.

[25] A. Thoma, A. Adel, M. Helmreich, T. Wehrle, F. Gramazio, and M. Kohler, "Robotic fabrication of bespoke timber frame modules," in *Proc. IEEE Robot. Fabr. Archit., Art Des.*, 2019, pp. 447–458.

[26] H. Fakhurldeen, F. Dailami, and A. G. Pipe, "CARA system architecture - A click and assemble robotic assembly system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 5830–5836.

[27] F. Chervinskii, S. Zobov, A. Rybnikov, D. Petrov, and K. Vendidandi, "Auto-Assembly: A framework for automated robotic assembly directly from CAD," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 11294–11300.

[28] "Fetchit! The mobile manipulation challenge." Accessed: Mar. 21, 2023. [Online]. Available: https://opensource.fetchrobotics.com/competition

[29] Y. Yokokohji et al., "Assembly Challenge: A robot competition of the Industrial Robotics Category, World Robot Summit–summary of the pre-competition in 2018," *Adv. Robot.*, vol. 33, no. 17, pp. 876–899, 2019.

[30] ICRA 2022, "Robotic grasping and manipulation competition: Manufacturing track," 2022. [Online]. Available: https://www.nist.gov/el/intelligent-systems-division-73500/icra-2022-robotic-grasping-and-manipulation-competition

[31] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.

[32] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and Z. Corse, "Local optimization for robust signed distance field collision," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 3, no. 1, pp. 1–17, 2020.

[33] M. Sridharan, M. Gelfond, S. Zhang, and J. L. Wyatt, "REBA: A refinement-based architecture for knowledge representation and reasoning in robotics," *J. Artif. Intell. Res.*, vol. 65, pp. 87–180, 2019.

[34] M. Gelfond and D. Inclezan, "Some properties of system descriptions of ALD," *J. Appl. Non-Classical Logics*, vol. 23, no. 1-2, pp. 105–120, 2013.

[35] M. Gelfond and Y. Kahl, *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach*. Cambridge, U.K.: Cambridge University Press, 2014.

[36] E. Balai, M. Gelfond, and Y. Zhang, "SPARC - Sorted ASP with consistency restoring rules," 2013, *arXiv:1301.1386*.

[37] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Multi-shot ASP solving with Clingo," *Theory Pract. Log. Program.*, vol. 19, pp. 27–82, 2019.

[38] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! Task constructor for task-level motion planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 190–196.

[39] S. R. Chhatpar and M. S. Branicky, "Search strategies for peg-in-hole assemblies with position uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 1465–1470.

[40] C. R. Garrett et al., "Integrated task and motion planning," *Annu. Rev. Control Robot., Auton. Syst.*, vol. 4, pp. 265–293, 2021.

[41] J. Yamada, C.-M. Hung, J. Collins, I. Havoutis, and I. Posner, "Leveraging scene embeddings for gradient-based motion planning in latent space," in *Proc. IEEE Int. Conf. Robot. Automat.*, London, United Kingdom, 2023, pp. 5674–5680, doi: 10.1109/ICRA48891.2023.10161427.

[42] J. Yamada, J. Collins, and I. Posner, "Efficient skill acquisition for complex manipulation tasks in obstructed environments," 2023, *arXiv:2303.03365*.

[43] J. Yamada et al., "Motion planner augmented reinforcement learning for obstructed environments," in *Proc. Conf. Robot Learn.*, 2020, pp. 589–603.

[44] M. Lambeta et al., "DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 3838–3845, Jul. 2020.

[45] Y. Tian et al., "Assemble them all: Physics-based planning for generalizable assembly by disassembly," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–11, 2022.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on December 18,2023 at 14:16:21 UTC from IEEE Xplore. Restrictions apply.