Delegate the Details: Flexible Assembly Constraint Authoring for Collaborative XR

Ben Yang* Xichen He* Charlie Zou* Jen-Shuo Liu* Barbara Tversky* Steven Feiner*

Columbia University

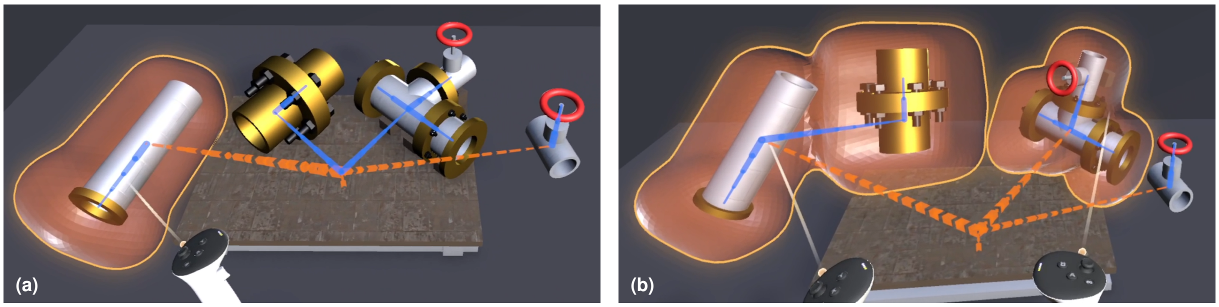

Figure 1: Staging parts on a pallet in our system (solid blue links show prescribed poses; dashed orange links show delegated poses). (a) Pallet children are a gray pipe and valve at delegated poses and a brass flanged pipe and T-pipe at prescribed poses. Pointing at the gray pipe displays its bounding hull. (b) The author has made the flanged pipe a prescribed child of the gray pipe, expanding its bounding hull, and has moved the T-pipe and toggled its pose to be delegated, pointing at it to show its bounding hull.

Abstract

Extended reality (XR) interfaces for assembly specification typically require users to specify the 6DoF pose of every part. This works well when exact poses are necessary, but can impose unneeded constraints when not. We present a collaborative XR constraint-authoring interface that supports a spectrum of specification authority. In our interface, an *author* can specify groups of parts and transfer them to a human or robot *executor* that carries out the author's instructions in the physical world. In *prescribed* mode, the author specifies the 6DoF pose of a part or group. In *delegated* mode, the author entrusts the executor to determine the 6DoF pose of a part or group. These modes can be mixed within a single task. We conducted a within-subjects study comparing prescribed-only authoring against dual-mode authoring. We found that dual-mode authoring significantly reduced authoring task completion time and perceived workload. Delegation rate increased with structure complexity. Most participants preferred dual-mode. We also evaluated both conditions offline with a simulated robot executor, feeding authored constraints through a workspace optimizer and motion planner. After optimization, the dual-mode workflow produced significantly greater space between groups to support physical assembly than prescribed-only specification.

Index Terms: Extended reality collaboration, human–robot interaction, assembly constraints, task specification.

1 Introduction

Extended reality (XR) interfaces for assembly task specification have evolved from low-level direct control to high-level goal specification [3, 62, 61]. Rather than manually guiding each action, users can specify desired part poses and let downstream motion planning systems [14] determine how to achieve them. This shift reduces operator workload and enables asynchronous workflows in which specification and execution proceed independently [50].

However, current goal-specification interfaces typically require users to define the 6DoF pose of every part or operation [33, 3, 30]. While straightforward for a single part, this approach is cumbersome for assembling multi-part hierarchical structures [66]. Consider a manufacturing assembly line where components are first assembled into subassemblies on dedicated fixtures before the loaded fixtures are combined into a final product: the user must specify each part's intermediate pose even though the essential constraints are which parts connect and how subassemblies relate to each other in the final product. Many of these intermediate pose decisions could be offloaded to a downstream executor (human or robot) that can optimize for feasibility, collision avoidance, and workspace utilization. This suggests a spectrum of specification authority: the user retains control where it matters and delegates the rest.

We present an XR constraint-authoring interface that supports such a spectrum of specification authority, ranging from full author prescription of part pose to delegation of determining part pose to an executor (Fig. 1). We use "specification authority" to mean the author's choice of how much to specify, as distinct from the conventional Human–Robot Interaction (HRI) sense of robot autonomy. We frame this as a separation between the *author*, who specifies constraints, and the *executor*, who carries them out. The executor may be a robot, a human worker, or any downstream system; the authored constraints are executor-agnostic. Our implementation and evaluation target robot execution, where the separation is especially valuable: a human author understands assembly intent but lacks knowledge of robot kinematics, while the robot's planner can optimize spatial layout for reachability and collision avoidance.

Our approach uses headset XR because it supports head tracking, stereoscopy, full-scale visualization, and bimanual interaction, all important for manipulating virtual parts in 3D assembly tasks. In contrast, conventional 3D CAD and robot-programming tools use mouse and keyboard on monoscopic desktop displays without head tracking, typically relying on unimanual point-and-click interaction to grab, move, and rotate parts in 3D and manual view control to look around. This would make our task of specifying poses and relationships of 3D parts more difficult and time consuming.

The author works directly with virtual models of the parts to be assembled. Our core operation is *bimanual nesting* (Sec. 3), where successive nesting forms a hierarchy of *groups* (Fig. 2). Biman-

*{by2297, xh2623, jz3331, jl5004, bt2158, feiner}@columbia.edu

ual nesting has no direct mouse-and-keyboard analogue: holding one part in each hand declares a parent–child relationship (Fig. 3). Each group is anchored by one part that defines the group's reference frame, may contain child groups, and is either *prescribed* (the author specifies the group's 6DoF pose relative to its parent group) or *delegated* (the executor determines that pose). Authors can mix these modes within one task. Delegation applies only where placement is flexible.

We conducted a user study that included both *assembly* (in which two parts are mated at exact relative poses, leaving nothing to delegate), and *arrangement* (in which pose can be delegated). We evaluate the effects of dual-mode authoring (in which a pose can be either prescribed or delegated) vs. prescribed-mode authoring on task time, inter-group clearance (the minimum distance between any two groups' bounding hulls, defined in Sec. 3.3), workload, and preference.

Our contributions include:

- An XR constraint-authoring interface supporting a spectrum of specification authority, from prescribed (author-specified poses) to delegated (executor-determined poses), with per-group modes mixed within one task.
- A within-subjects study ($n = 20$) that compares prescribed-only against dual-mode authoring across three hierarchical structures, tests whether dual-mode authoring reduces task completion time and perceived workload, and analyzes how structural complexity affects efficiency.
- An offline end-to-end workflow evaluation measuring inter-group clearance after workspace optimization.

2 Related Work

We situate our work at the intersection of constraint specification for collaborative assembly and assembly feasibility evaluation.

2.1 Assembly Constraint Specification

Assembly task specification spans a spectrum from full automation to full human control.

Fully automated specification. At one end, automated approaches derive constraints entirely from CAD models, structured data, or learned demonstrations. Task and motion planning integrates symbolic and geometric planning [69], with recent work scaling to hundreds of robots [8]. Constraint programming jointly optimizes workspace layout and task scheduling [63, 12], and automated cell configuration systems derive robot placements from product models [4]. Vision–Language models (VLMs) extract specifications from instruction manuals [56, 55], and approaches based on Large Language Models (LLMs) generate assembly plans from task descriptions [2]. These systems assume a fixed assembly goal and extract rather than create a specification.

Full human control. At the other end, the user makes every spatial decision, whether at the trajectory level or, as in goal-based teleoperation interfaces, as desired end states [62, 33]. Aoyama et al. [3] extended this to asynchronous goal assignment where users specify 6DoF assembly targets in XR, supporting concurrent specification and execution. Other end-user authoring environments enable non-experts to program workflows through workflow authoring [47], embodied XR demonstration [11], situated programming [49], multimodal XR with LLM-based composition [25, 24], XR skill-parameter visualization with digital-twin validation [30], and hybrid XR authoring for HRI [35]. XR assembly interfaces have been evaluated for CAD modeling [15] and collaborative tasks [46]. Yet these interfaces still require the author to fully specify the target configuration; none offers a way to delegate pose decisions to an executor.

Mixed-initiative approaches. Between these extremes, shared-control approaches blend user input with autonomous capabilities [48, 64]. Senft et al. [50] introduced task-level authoring for variable robot autonomy, Hagenow et al. [21] combined end-user programming with shared autonomy, and Porfirio et al. [42] demonstrated goal-oriented programming at variable granularity. Hierarchical planning frameworks decompose collaborative assembly into task allocation between human and robot [26]. These approaches allocate autonomy over whole tasks or actions: even pose determination for a set of parts would be assigned to the human or the robot wholesale. We instead split that authority within a task, pose by pose.

Direct precedents. Yang et al. [65, 66] proposed within-group specification modes for hierarchical assembly in XR, letting users choose between full pose specification and delegation at each level of the assembly tree (termed "absolute" and "relative", which we reframe as "prescribed" and "delegated"). We build on this by adding an implemented authoring interface, a user study, and a downstream planner evaluation. Aoyama et al.'s system [3] is the closest prior work to our prescribed-only condition: it supports 6DoF goal-pose specification for individual parts in XR with asynchronous execution, but requires the user to specify every pose and offers no delegation mechanism. In contrast to these alternatives, fully prescribed XR specification on one side and fully automated extraction on the other, our work introduces *constraint-based* goal specification that mixes prescribed and delegated modes group by group: rather than demonstrating a single desired configuration, the author specifies which spatial relationships are required while delegating the remaining pose decisions to the executor. To our knowledge, ours is the first XR assembly system to combine per-group prescribed/delegated authoring with a closed-loop evaluation that feeds the authored specifications through a workspace optimizer and a simulated robot.

XR interaction techniques. Our interface uses bimanual nesting to build the constraint hierarchy. Bimanual interaction in XR builds on Guiard's account of asymmetric hand roles, in which the nondominant hand sets the frame of reference within which the dominant hand acts [20]; techniques span proxy-based two-handed manipulation [40, 41, 67], miniature workspaces [52], and extended reach [43]. Recent work has combined bimanual input with pointing [68], gaze [36], and curve sketching [32]. Object manipulation spans direct grab [29], near-field [31], and hybrid approaches [5]. Our bimanual nesting technique differs in purpose: in addition to manipulating parts, simultaneously holding two parts with both hands *creates a semantic parent–child relationship* in the constraint hierarchy. None of these prior techniques combines hierarchy construction with per-group authority over pose specification.

2.2 Assembly Feasibility Evaluation

Evaluating assembly feasibility determines whether a given assembly specification can be physically executed by a robot, checking that target poses are collision-free, stably supported, and within the manipulator's kinematic reach. This problem is well-studied at the planning level. ASAP [54] generates physically feasible assembly sequences by ensuring intermediate physical stability and realizability. Inverse reachability maps [58] identify feasible robot base placements from which target poses are reachable. These tools assume a fixed target specification and evaluate whether a robot can execute it. They do not address how the specification was created or whether the specification method affects execution outcomes.

The distinction between prescribed and delegated constraints has significant implications for planning success. Prescribed constraints fix part poses, potentially leading to infeasible configurations due to collisions or kinematic limits. Delegated constraints maintain inter-part relationships while allowing the planner to optimize the delegated poses throughout the task, intermediate config-

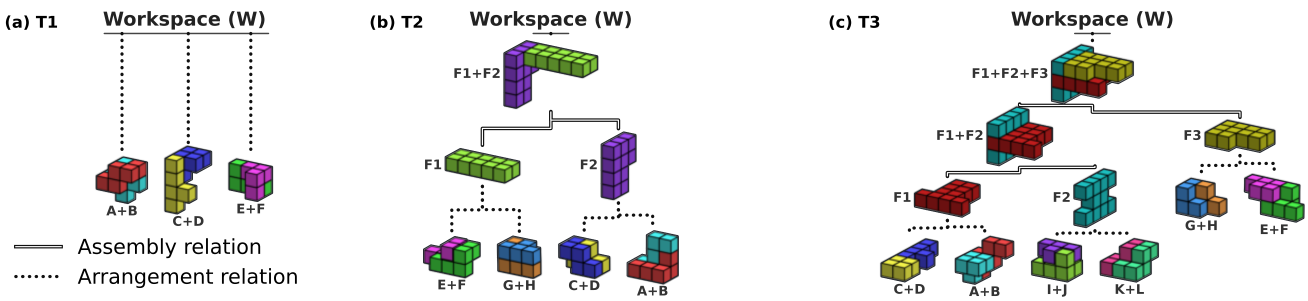

Figure 2: Study task hierarchies. Black double-stroke edges denote assembly relations; black dotted edges denote arrangement relations. Leaf labels (e.g., “A+B”) denote two-part subassemblies; their constituent part nodes are omitted for simplicity. Each hierarchy is rooted at workspace W. (a) T1 has no fixtures. (b) T2 joins two fixtures (F1 and F2). (c) T3 joins three fixtures (F1, F2, and F3) in two stages. Constraint mode is not shown: assembly relations are prescribed, while arrangement relations may be prescribed or delegated.

urations included, thus expanding the feasible solution space. We adopt established feasibility evaluation methods [54, 58] and apply them to compare how the specification paradigm affects execution outcomes, which has not been addressed in prior work.

3 System Description

Our interface organizes parts into a hierarchy, visualizes constraint structure through animated directional lines and on-demand bounding hulls, and exports specifications for execution.

3.1 Hierarchical Constraint Model

An assembly is represented as a tree of *groups* (Fig. 2). This mirrors the standard decomposition of a product into a hierarchy of subassemblies that underlies classical assembly-sequence planning [23], in which each node is a subassembly composed of its children. Each group is anchored by a single part that defines the group’s reference frame, and may have any number of child groups nested beneath it; each child group expresses its pose relative to its parent’s reference frame. The root group represents the workspace (e.g., an assembly table). Which part anchors a group’s frame does not affect the resolved assembly, since all poses are expressed relatively; any residual freedom in a group’s placement is resolved explicitly by its constraint mode.

Each group carries a constraint mode that can be independently set to *prescribed* or *delegated*:

- **Prescribed**: The author specifies the 6DoF pose of this group relative to its parent. The executor will check to see if it can reproduce it exactly; a prescribed pose that cannot be executed throws an error since the specification is infeasible.

- **Delegated**: The author entrusts the executor to determine the 6DoF pose of this group relative to its parent, subject to feasibility constraints (collision avoidance, support, reachability).

The mode describes a group’s relationship to its parent, not its children. Sibling groups under the same parent can have different modes. This enables fine-grained control within a single group: for example, two subassemblies on a shared fixture might have different delegation states relative to the fixture, while the fixture itself is either prescribed or delegated relative to its parent. Delegation is also the model’s ambiguity-resolution mechanism: when the author does not commit to an exact placement, the delegated mode assigns specifying the group’s residual degrees of freedom to an optimizer, resolving any ambiguity.

3.2 Interaction Design

Authors build the constraint hierarchy through the following mechanisms:

Bimanual nesting. To establish parent–child relationships (e.g., grouping subassemblies under a shared parent), the author holds

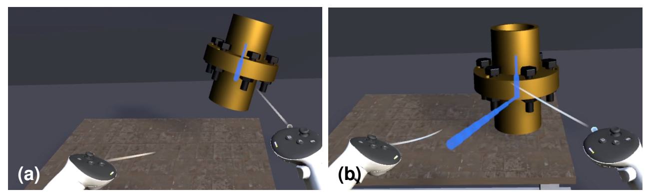

Figure 3: Bimanual nesting. (a) The author holds the flanged pipe and the pallet, one per hand; a blue link marks the prescribed connection between the two parts of the flanged pipe. (b) Releasing the flanged pipe nests it under the pallet in prescribed mode (blue link).

one part with each controller and releases the intended child, as shown in Fig. 3. The released part’s group is reparented under the held part’s group. This two-handed action creates a semantic parent–child relationship distinct from physical snapping: it defines which subassemblies belong together in the constraint tree without requiring geometric contact. Reparenting to the world is performed by holding the child and the workspace root, then releasing the child. We use a two-handed action because each parent–child link joins one parent and one child: one part per hand, in line with asymmetric bimanual interaction [20]. This avoids the menu navigation that a unimanual or modal alternative would require. Controller buttons keep toggling and confirmation available while both hands hold parts, without redirecting gaze to a menu.

Cycle correction. If reparenting a group under one of its own descendants would create a cycle, the interface auto-corrects: the descendant is first promoted (reparented to the group’s current parent) before the original reparenting proceeds. Both operations form one undo entry, so one button press reverts them.

Undoing nesting. Pressing either thumbstick reverts the most recent nesting operation from a global stack. For cyclic auto-corrections, the compound entry ensures that both the promotion and the reparenting are reverted together.

Within-group delegation toggle. While holding a part, the author presses a designated controller button to toggle that group between prescribed and delegated modes. Toggling to delegated means the optimizer will determine the group’s pose relative to its parent at confirmation time.

Confirmation. Pressing the designated confirm button while holding a part confirms that group and its changed children. Pressing confirm with nothing held performs a global confirm across all confirmable groups. Confirmation commits the group’s current constraints for export; the validity checks applied at confirmation in this paper are part of the user study configuration (Sec. 4.2).

3.3 Visualization

Two complementary visual encodings communicate the constraint hierarchy and delegation state (Fig. 1):

Animated directional lines connect each child group to its parent using chevrons that flow parent-to-child, coded solid blue (prescribed) or dashed orange (delegated). We chose animated directional edges following Büschel et al. [10], who found that animated edges were effective for directed graph perception in XR, and Joos et al. [27], who noted that node-link representations outperform matrix encodings for network comparison tasks in XR. Lines are always visible, providing a persistent hierarchy overview; only delegated groups draw a link to the workspace root. Color and motion encode orthogonal channels: the animated chevrons convey *direction* (parent-to-child) and hue (blue/orange) conveys *mode* (prescribed/delegated). Hue is a preattentive cue that does not require tracing a line to perceive in dense 3D scenes; participants were introduced to this mapping during training (Sec. 4.6).

Demand-driven bounding hulls appear around delegated groups and their descendants only when a controller is within 12 cm of the group's centroid or points at the group, indicating that the executor will determine the group's 6DoF pose. Each bounding hull is an implicit isosurface [7]: per-part voxel centers contribute sphere signed-distance-field (SDF) primitives blended via polynomial smooth-min [44], with the mesh extracted by GPU marching cubes [34]. The resulting surface conforms organically to the group's geometry rather than enclosing it in a convex bounding volume, providing a tighter visual approximation of group extent. This visualization avoids clutter by combining persistent lines for hierarchy overview with on-demand hulls for group membership, following evidence that multi-layout designs outperform single representations for hierarchical visualization in immersive environments [6]. Prescribed groups are never bounded by hulls, so hulls are a visual marker of delegation.

3.4 End-to-End Pipeline

The interface exports each confirmed specification to a downstream execution pipeline; our implementation targets a simulated robot executor, exercised by the offline evaluation (Sec. 6).

XR Frontend (Unity): Authors specify constraints using the interface described above. The interface records part poses, hierarchy structure, and within-group constraint modes.

Workspace Optimization: For delegated groups, the optimizer solves a constrained nonlinear program whose only decision variables are the anchor poses of the delegated groups (each a 6DoF pose). Prescribed poses are held fixed and pass through directly. Collision-freeness and support are enforced as *hard* constraints: a minimum-separation bound between every pair of group bounding hulls and a resting-surface floor on each group's height, alongside workspace-bound constraints. Subject to these, the optimizer minimizes a soft objective: adherence to the authored relative poses, a resting-contact term, reachability against the manipulator's inverse-reachability map [58], and lower-weighted manipulability and joint-regularity terms. We solve it with Drake [53], using SNOPT [17] (falling back to IPOPT [60]), seeded from the authored layout, and poses of non-anchor parts followed by breadth-first constraint propagation through the hierarchy ($T_{\text{obj}} = T_{\text{ref}} T_{\text{rel}}$).

Motion Planning: Optimized poses are translated into MoveIt [14] planning goals using RRT-Connect (OMPL, 2s timeout), sequenced by Task Constructor [19].

Simulation: Trajectories are executed on a simulated Universal Robots UR5 arm in MuJoCo [57] to verify end-to-end feasibility.

4 User Study

4.1 Pilot Study

We conducted an initial pilot study ($n = 4$), approved by our institutional review board, to refine the interface design and calibrate the study parameters. The four pilot participants were drawn from our institutional community and had prior XR experience comparable to our target population. They completed the full protocol and gave verbal feedback.

Visualization. Our initial interface used nested bounding hulls for both prescribed and delegated hierarchy. However, pilot participants reported excessive visual clutter in deeper hierarchies. We replaced these with the current two-layer design: animated directional lines for persistent hierarchy encoding and bounding hulls appearing only on delegated groups when the controller is nearby or pointing at them (Sec. 3).

Interaction. Pilot sessions revealed that participants frequently attempted proximity-based nesting rather than the bimanual trigger action. We refined the visual and haptic feedback to make the bimanual nesting action more discoverable and added scaffolded practice trials to the training protocol.

Study calibration. Pilot-study data informed the power analysis and sample-size determination (Sec. 4.5).

4.2 Formal Study

Design summary. The study used a within-subjects (repeated-measures) design with one manipulated independent variable, *instruction mode* (prescribed-only vs. dual-mode), and task structure (T1/T2/T3) as a within-subjects complexity factor. Dependent variables were task completion time (primary), unweighted NASA-TLX workload, forced-choice preference, and offline robot-planning clearance. Each participant completed both conditions in counterbalanced order with condition-specific training before each block, and trial order within a block was semi-randomized (T3 never first because of its complexity). We did not include a delegation-only condition because assembly relationships mate parts at exact relative poses, leaving no freedom to delegate; only arrangement relationships are delegable. In dual-mode, participants could therefore delegate every delegable group. Removing the assembly level to allow only delegation would eliminate the need for human authorship, representing fully automatable design tasks that we are not trying to address.

We conducted a within-subjects study, approved by our institutional review board, comparing prescribed-only authoring against dual-mode (prescribed and delegated) authoring for hierarchical assembly constraint specification. To compare access to delegation while holding other mechanics constant, we adapted the full interface described above as follows:

Assembly mechanics. All study trials use *polycube pieces* to control for part geometry, with connection points defined in trial configuration files: pentacubes (five unit cubes) for regular parts and decacubes (ten unit cubes) for fixtures, providing uniform geometric complexity with full 6DoF relevance.

Parts snapped together when brought within ± 2 cm translational and $\pm 10°$ rotational tolerance of defined connection points (informed by mid-air docking thresholds reported by Vuibert et al. [59], adjusted for our polycube geometry), automatically establishing the snapped part as a child of the stationary target. Children followed their parent when moved, forming rigid units. All parts and groups had to rest on a supporting surface (a fixture or the table); confirmation was rejected while parts collided or lacked support, with visual and haptic feedback identifying the failure.

Task guidance. Guidance was provided through floating World-in-Miniature (WiM) reference displays [52], as shown in Fig. 4: Each task received its own WiM depicting the target configuration at $0.3\times$ scale by default. Each assembly WiM is enclosed in a thin white bounding box to make its extent easy to see against the scene; each arrangement WiM is instead enclosed in a teal wireframe sphere, linked by tapered teal lines to the WiMs it arranges; pointing at a WiM draws leader lines to its parts. Participants could grab a WiM to reposition or rotate it and could rescale it with the thumbstick (0.15–$1.0\times$). These WiMs served as read-only guidance displays and could not manipulate parts they represented, unlike many other WiMs [52]. WiMs were revealed progressively as dependency-graph prerequisites were satisfied, so in-

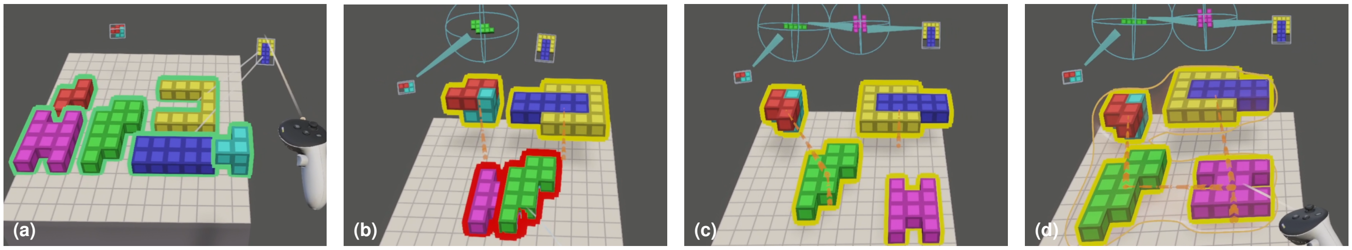

Figure 4: Nested-fixture practice trial with WiM guidance. (a) Starting poses of the parts, with the initial guidance WiMs at the upper left for the red–cyan (RC) subassembly and upper right for the blue–yellow (BY) subassembly; all outlines are green (unchanged since the last confirmation). Two lines connect the BY WiM parts to the physical BY parts since the controller is hovering over the WiM. (b) When RC and BY are assembled and delegated to the workspace root, the green fixture's WiM is revealed; its tapered line marks where RC must be arranged. The overlapping green and magenta fixtures are outlined in red (colliding); RC and BY are outlined in gold (changed). (c) When RC is delegated onto the green fixture, and the fixture is delegated onto the table, the final WiM is revealed, showing that the green fixture and BY must be arranged on the magenta fixture; all outlines are now gold. (d) All tasks have been satisfied and pointing a controller at the magenta fixture displays a bounding hull around the whole delegated group; all outlines are gold.

dependent branches could progress at different rates; WiMs were initially placed along the near edge of a virtual table and consolidated as merges occurred. A persistent, world-locked control panel shown throughout each trial displayed the control mappings and a legend for the blue/orange (prescribed/delegated) and confirmation-state colors, with content adapted to the condition. A help button pulled up a key of these encodings on demand.

Offline evaluation. No optimizer or planner ran during the study. The interface recorded constraint specifications at each confirmation action; these were replayed offline to evaluate planning feasibility (Sec. 6) under identical runtime conditions.

Conditions. Both conditions share all adaptations above, the same hierarchy visualization (animated directional lines), bimanual nesting, confirmation mechanics, and per-part outline feedback for confirmed (green), changed since last confirmation (gold), and colliding (red) states, shown in Fig. 4. In the *prescribed-only* condition, the delegation toggle is locked (all groups remain prescribed, no bounding hulls appear); participants specify all poses themselves. In the *dual-mode* condition, all groups begin as prescribed; the toggle is unlocked and bounding hulls provide feedback when participants mark arrangement-level groups for executor-determined placement. The treatment therefore varies access to delegation and its bounding-hull feedback; all other interaction and confirmation mechanics are shared.

4.3 Research Questions and Hypotheses

All study tasks are fundamentally constraint-specification acts: placing one entity relative to another. We distinguish two task types based on the constraint level:

- **Assembly tasks** connect parts face-to-face with a defined target pose (tight tolerance). These are mechanically identical in both conditions; the optimizer has no role.
- **Arrangement tasks** lay out assembled groups on a fixture surface (bounded, no target pose, collision-free and supported). Dual-mode applies delegation and optimization here.

We investigated four research questions:

- **RQ1 (Task Time):** Does dual-mode authoring reduce task completion time compared to prescribed-only authoring, and does the effect differ by task type?
- **RQ2 (Robot Outcomes):** Does the end-to-end dual-mode workflow produce greater inter-group clearance after optimization than prescribed-only authoring?
- **RQ3 (Workload):** Does dual-mode authoring reduce perceived workload?
- **RQ4 (Preference):** Do authors prefer dual-mode?

Confirmatory hypotheses ($\alpha = .05$ each):

- **H1 (Task Time):** *Dual-mode authoring reduces task completion time* (main effect of condition in Gamma Generalized Linear Mixed Effects Model (GLMM) with log link and task_type as covariate). The Instruction Mode × Task Type interaction is reported as an exploratory contrast.
- **H2 (Robot Outcomes):** *The dual-mode workflow produces greater inter-group clearance after optimization than prescribed-only authoring* (paired Wilcoxon signed-rank test on participant means).
- **H3 (Workload):** *Unweighted NASA-TLX total score is lower in dual-mode than prescribed-only* (paired Wilcoxon signed-rank).
- **H4 (Preference):** *Participants prefer dual-mode in post-study forced-choice* (binomial test) with a reasoning item (speed, effort, trust, or other).

Exploratory measurements: Instruction Mode × Task Type interaction (whether the dual-mode advantage differs between assembly and arrangement), early/late delegation timing, hierarchy churn, per-structure breakdowns, MRT-A as moderator, and trial order effects.

4.4 Task Design

All trials use pick-and-place assembly; fastening operations are excluded as orthogonal to specification strategy. All assembled groups must be physically supported (Fig. 4), so participants must manage placement throughout assembly, not only at explicit arrangement levels. Each structure also includes a final table-arrangement step (placing the completed fixture assembly on the table); this step is trivially short (typically <1 s) because it involves no meaningful spatial decision.

We designed three structures of increasing complexity, spanning two to five hierarchy levels, inclusive of the workspace W (Fig. 2):

- **T1** (6 parts, 2 levels, 0 fixtures): Three pair assemblies on the table (3 assembly + 1 arrangement = 4 tasks).
- **T2** (10 parts, 4 levels, 2 fixtures): Assemblies on two fixtures, followed by fixture connection and table placement (5 assembly + 3 arrangement = 8 tasks).
- **T3** (15 parts, 5 levels, 3 fixtures): Three-fixture assembly with multi-level connections (8 assembly + 4 arrangement = 12 tasks).

These three structures vary in number of parts (6, 10, 15), fixture count (0, 2, 3), and hierarchy depth. Structure is treated as an ordinal complexity factor in the analysis since the dimensions cannot

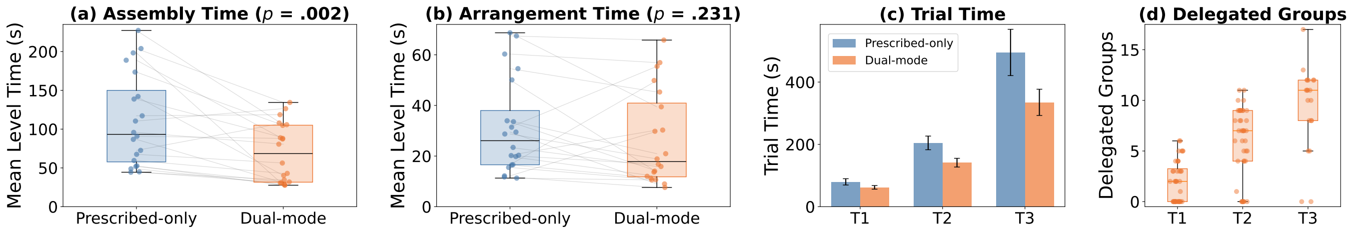

Figure 5: Task time and delegation. (a) Assembly time by condition (paired lines per participant; $p = .002$). (b) Arrangement time ($p = .23$, n.s.). (c) Trial time by structure complexity (± 1 SE). (d) Delegated groups per trial ($H = 42.3$, $p < .0001$).

be disentangled. Per-structure results show how efficiency varies across these structures.

Trial set. Each block contains five scored trials: 2×T1, 2×T2, 1×T3. Trial order is semi-randomized (T3 cannot be first, as its 15-part complexity could cause early fatigue, contaminating subsequent trials). The same polycube configurations are reused across blocks, controlling for geometry; any performance difference is attributable to the condition. Using the same configurations across blocks also reduces the asset-creation burden.

The WiM-based guidance (per-task miniatures with progressive reveal gated by the dependency graph) and all assembly mechanics are shared across conditions. Neither condition provides ghost parts or directional cues; participants reason spatially from the WiMs.

A *level-task* comprises all the work at one hierarchy level of one trial. Each trial produces 2 (T1), 4 (T2), or 5 (T3) level-task observations (one per hierarchy level); after excluding zero-duration observations, 588 level-task observations enter the primary analysis.

4.5 Participants and Power Analysis

We determined the target sample size using data from our initial pilot study (Sec. 4.1). The primary target was H1 (main effect of condition on task completion time). Pilot participants showed a paired effect size of $d_z = 1.47$ on trial time (log-scale $d_z = 1.47$, ratio $= 1.48\times$). Conservatively halving this effect to $d_z = 0.74$ to account for regression to the mean and the simplification from a GLMM to a paired t-test, power analysis using the `pwr` R package [9] indicated $\geq 80\%$ power at $n = 17$ ($\alpha = .05$, two-tailed). We targeted $n = 20$ to provide a buffer for exclusions. At $n = 20$, sensitivity analysis indicated a minimum detectable effect of $d_z = 0.66$ at 80% power, and the achieved power for the conservative effect size was 87.6%. This paired t-test estimate is conservative relative to the Gamma GLMM used in the primary analysis, which leverages level-task observations rather than participant-level means.

We recruited 21 participants from our institutional community (identified below as P1–P21, with one excluded for failure to complete either practice block), none participating in the pilot study (Sec. 4.1), yielding a final sample of $n = 20$ (9 female, 11 male; ages 19–33, $M = 23.6$, $SD = 3.1$). Nine reported monthly XR use, two weekly use, seven daily or more, and two no prior XR experience. All participants had normal or corrected-to-normal vision, passed Ishihara color-vision–deficiency screening, and completed the Stereo Fly test [51] and Mental Rotation Test (MRT-A) [39] as pre-questionnaires. Exclusion criteria included: cybersickness requiring session termination and failure to complete either practice block. Participants received modest monetary compensation of 15 USD; all sessions were completed within 90 minutes.

4.6 Procedure

Participants used a Meta Quest 3 headset with two controllers, connected through Meta Quest Link to a Windows 11 computer with an NVIDIA RTX 3080 GPU. After providing informed consent, participants completed a demographics questionnaire and the tests mentioned above. The experimenter explained the task and demonstrated the interactions relevant to the first block.

After pre-study screening, participants completed two counterbalanced blocks. Each block began with condition-specific training (three scaffolded practice trials: 2-part assembly basics, 4-part arrangement with delegation, and a 6-part fixture workflow; dual-mode blocks added a non-interactive walkthrough of nesting and delegation toggle). The scored block comprised five trials (2×T1, 2×T2, 1×T3), followed by an unweighted NASA-TLX questionnaire [37], modified to use a 1–7 scale [1], with 1 as best, rather than the original 0–20 scale. After a break, participants completed the second block under the other condition (same configurations, new order). A forced-choice preference item followed.

4.7 Measures

Each measure below is tagged with its research question, hypothesis, and analysis; delegation and toggle counts are analyzed with Kruskal–Wallis tests, and the remaining additional measures are exploratory and descriptive.

Task completion time (primary dependent variable; RQ1/H1, Gamma GLMM with log link): Time from when a level becomes active (all prerequisites in the dependency graph satisfied) to when all groups at that level are confirmed. Levels are contiguous and non-overlapping: the completion event for one level triggers the start of the next level. Assembly levels end at the final snap confirmation; surface placement of assembled groups occurs in the subsequent arrangement level.

Unweighted NASA-TLX (RQ3/H3, Wilcoxon signed-rank): Six subscales adapted to 7-point Likert scales (1–7) [22], administered once per block.

Preference (RQ4/H4, binomial test): Post-study forced-choice with reasoning item (speed, effort, trust, or other).

Planner metrics (RQ2/H2, Wilcoxon signed-rank): Offline inter-group clearance (Sec. 6).

Additional measures included arrangement-task grabs, dominant-hand path length from 30 Hz tracking, delegation and toggle counts (dual-mode only), MRT-A score, and interviews.

5 Results

5.1 Analysis Plan

Each confirmatory hypothesis tests a different construct and is analyzed independently at $\alpha = .05$. Type I errors are therefore controlled per comparison: the four confirmatory hypotheses are prespecified, test-distinct constructs, and are each tested exactly once, so no cross-family correction is applied; the six NASA-TLX subscales are treated as one follow-up family and Holm–Bonferroni corrected; all remaining analyses are labeled exploratory or descriptive and make no confirmatory claims. The **H1 model** is a Gamma GLMM with log link, which models positive right-skewed completion times with variance proportional to the mean:

```
task_time ~ condition + task_type + (1 |
participant) + (1 | trial),
```

where `condition` is sum-coded (± 0.5), `task_type` is binary (assembly vs. arrangement), and `trial` is a crossed random intercept (5 levels). Block order (which condition was experienced first)

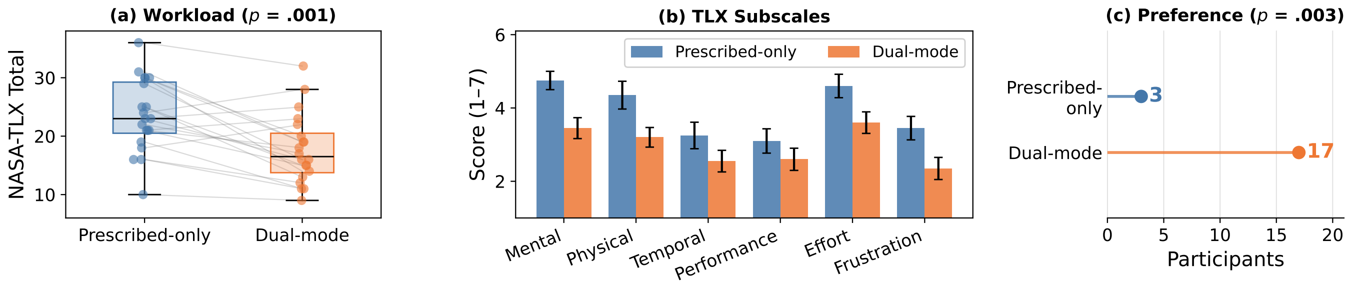

Figure 6: Workload and preference. (a) NASA-TLX totals by condition (paired lines per participant). (b) TLX subscale means (± 1 SE). (c) Forced-choice preference (17/20 dual-mode; $p = .003$).

was tested as a covariate to check for order and learning effects; it was not significant ($p = .24$) and was dropped. Coefficients on the log link are interpretable as multiplicative effects on completion time. Random slopes for condition were attempted but produced a singular fit (slope variance estimated at zero), indicating negligible between-participant variation in the condition effect; they were dropped for the more parsimonious random-intercepts model.

5.2 Task Time (RQ1, H1)

To address RQ1, we fit a Gamma GLMM with log link, condition (sum-coded) and task_type as fixed effects, and random intercepts for participant and trial ($n = 20$, 588 level-task observations). Dual-mode authoring significantly reduced task completion time: $b = -0.38$ on the log scale ($\exp(b) = 0.68$, a 32% multiplicative reduction), $z = -3.12$, $p = .002$. At the participant level, the overall paired effect on log trial time was $d_z = 0.67$, within the study's detectable range (minimum detectable effect $d_z = 0.66$; Sec. 4.5). In exploratory per-task-type contrasts, assembly tasks were significantly faster in dual-mode (prescribed-only $M = 112.2$ s vs. dual-mode $M = 68.5$ s; paired $t(19) = 3.59$, $p = .002$, Cohen's $d_z = 0.80$; Fig. 5). Arrangement tasks showed a non-significant 14.7% reduction ($M = 31.3$ s vs. $M = 26.7$ s; $t(19) = 1.24$, $p = .23$). The Instruction Mode $\times$ Task Type interaction was not significant ($b = 0.16$, $z = 0.68$, $p = .50$). The assembly-level speedup is concentrated at fixture-connection levels (L2/L3 in T2 and T3), where the effect is 2.5–2.7$\times$; the driving mechanism (fixture connection displacing the arranged subassemblies, which prescribed-only confirmation must repair while delegation absorbs it) is analyzed in the Discussion. Initial assembly levels (L0) show a smaller difference (1.4$\times$), consistent with the part-to-part assembly mechanics being identical across conditions.

In exploratory per-structure contrasts, the dual-mode advantage grew with structure complexity (T1: 22%, T2: 31%, T3: 36% faster; Fig. 5a), though the condition $\times$ structure interactions were not significant (T2: $b = -0.20$, $z = -1.67$, $p = .095$; T3: $b = -0.23$, $z = -1.57$, $p = .116$).

Robot planning outcomes (H2) are reported in Sec. 6.

5.3 Workload and Preference (RQ3, H3, RQ4, H4)

H3 (Workload). Unweighted NASA-TLX total scores (7-point adapted scales) were significantly lower in dual-mode ($M = 17.8$, $Mdn = 16.5$) than prescribed-only ($M = 23.5$, $Mdn = 23.0$; Wilcoxon $W = 192$, $p = .001$, $r = .72$). Five of six subscales reached significance: Mental ($p = .001$), Physical ($p = .004$), Temporal ($p = .014$), Effort ($p = .005$), and Frustration ($p = .006$); Performance was marginal ($p = .051$). All five survive Holm–Bonferroni correction across the six-subscale family (adjusted $p \leq .028$; $r = .66$–$.79$). All six favored dual-mode, confirming that delegation reduced perceived workload (Fig. 6a,b).

H4 (Preference). 17 of 20 participants (85%) preferred dual-mode (binomial $p = .003$, 95% CI 62–97%; Fig. 6c). The three participants who preferred prescribed-only cited familiarity (P8, P14) or discomfort with the additional interface complexity (P3).

5.4 Additional Analyses

Delegation rate. Delegation scaled significantly with structure complexity: $M = 2.0$ (T1), $M = 6.0$ (T2), $M = 9.4$ (T3) delegated groups per trial (Kruskal–Wallis $H = 42.3$, $p < .0001$, $\varepsilon^2 = .42$). Toggle usage followed the same pattern ($M = 1.7$, $M = 5.9$, $M = 9.4$ for T1, T2, T3 respectively). The overall delegation rate was 73.3%, scaling from 57.5% (T1) to 80.8% (T2) to 90.0% (T3). 19 of 20 participants delegated at least one group; one participant (P20) never delegated in dual-mode but is retained under intent-to-treat.

Grab count. Grab counts at arrangement tasks did not differ significantly between conditions (prescribed-only $M = 7.6$ vs. dual-mode $M = 9.0$; $U = 6217$, $p = .18$). The numerical increase in dual-mode is consistent with delegation shifting effort from spatial reasoning to physical exploration, but this interpretation remains tentative.

Hand path length. Dominant-hand path length was 20.6% shorter in dual-mode ($M = 31.0$ m vs. prescribed-only $M = 39.0$ m), though this did not reach significance ($t(19) = 1.30$, $p = .21$). The reduction was largest for T3 ($M = 32.9$ m vs. $M = 59.4$ m); neither the overall nor per-structure differences were statistically reliable.

5.5 Sensitivity Analyses

To assess robustness, we applied an extended Tukey fence ($\pm 3\times$IQR) to the level-task completion times, removing 11 extreme observations (all >289 s, primarily T3 assembly tasks in prescribed-only, where connecting fixtures displaced subassemblies and triggered repeated confirm failures). The H1 main effect of condition remained significant ($b = -0.31$, $z = -3.38$, $p < .001$). The same criterion applied to unweighted NASA-TLX paired differences identified no outliers (H3 $p = .001$ unchanged). We also verified robustness to participant exclusion: one participant (P20) never delegated in the dual-mode condition (intent-to-treat); excluding P20 strengthened the condition effect ($b = -0.35$, $z = -3.73$, $p < .001$) and preference remained significant (16/19, $p = .004$). All confirmatory results remain robust.

5.6 Qualitative Observations

Post-block free-response comments and the forced-choice preference reasoning item revealed three recurring themes.

Reduced spatial planning burden. The most frequent theme was that delegation removed the need to mentally plan table arrangements. P21 observed that dual-mode was "easier in the sense that I do not have to put stuff on the table every time I complete a task," adding that "it is easier to put all the assembly in the air as there are more room to manage." P5 reported the experience as "a lot more smoother and flexible," and P1 explained their preference by noting "it was easier to not have to put things together if it wasn't necessary." P7 cited that "a lot of difficulty was when I had to place it on the bottom surface and when I had to stack them together," which delegation removed. P5 specifically attributed their preference to not having "to worry about support."

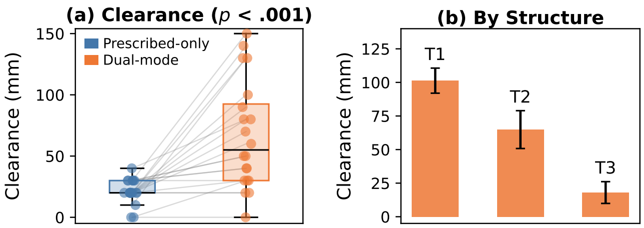

Figure 7: Inter-group clearance. (a) Minimum clearance per participant by condition (paired lines; $W = 0$, $p < .001$; 16/20 improved). (b) Clearance by structure complexity (± 1 SE).

Organizational affordances. Several participants used the hierarchical grouping system beyond its intended purpose. P8 used the parenting system "for organization rather than disabling the collider," suggesting emergent uses beyond our design intent. P9 valued the visual feedback, noting "being able to have pieces farther away but still connected through a parent child relationship made it easier to visualize," and that the hull display helped manage spatial relationships. P11 said the "toggling system helped" manage many parts with "less effort in moving around individual things."

Learning curve and interface complexity. The three participants who preferred prescribed-only (P3, P8, P14) cited interface complexity as the primary reason. P14 reported sometimes forgetting "the meaning of the blue/orange lines (what color represents what mode)" and preferred prescribed-only because there was "one less button to think about." P8, who stated they "did not really use the orange frame functionality" initially, preferred the simpler UI. However, participants who initially struggled reported rapid adaptation: P6 noted the dual-mode interface "got a lot easier" with practice, and P16 was frustrated in prescribed-only because they "couldn't use the collision / support avoider" from dual-mode.

6 Robot Planning Evaluation (RQ2, H2)

To address RQ2 and test H2, we conducted an offline planner evaluation comparing specifications from both conditions (Fig. 7).

6.1 Method

We replayed each participant's constraint specifications offline through the workspace optimizer and motion planner. Assembly-level poses are identical between conditions; the difference arises at arrangement levels, where dual-mode uses optimizer-chosen poses for delegated groups. We tested 100 specifications per condition (5 trials $\times$ 20 participants), evaluating inter-group clearance, which determines whether a robot gripper can access each part without contacting adjacent groups. To validate end-to-end feasibility, we executed all specifications through a simulated UR5 and gripper in MuJoCo [57] with collision-free motion planning (RRT-Connect, 2 s timeout, 8 grasp orientations).

6.2 Results

Participants delegated at least one group in 69 of the 100 dual-mode trials (69%). The hierarchical workspace optimizer succeeded on all 69 trials (100% feasibility), resolving poses in a mean of 0.02 seconds per trial. In the end-to-end execution check, the simulated UR5 and gripper completed pick-and-place sequences for all 200 specifications (100 per condition), with comparable execution times across conditions (prescribed-only: $M = 125$ s; dual-mode: $M = 132$ s; $p = .19$).

Clearance. Prescribed-only specifications had a median clearance of 0 mm (mean 21.5 mm), indicating that participants frequently placed groups flush against each other or overlapping. After optimization, dual-mode specifications achieved a median clearance of 50 mm (mean 67.0 mm), a significant improvement ($W = 0$, $p < .001$, $r = .87$, Wilcoxon signed-rank on paired per-participant means; 16 of 20 participants improved). Clearance scaled inversely with structure complexity: T1 trials (table-only) averaged 101 mm, T2 (2 fixtures) 65 mm, and T3 (3 fixtures) 18 mm; for reference, the simulated gripper's maximum opening is 85 mm. A robot gripper must approach and grasp each part without colliding with neighboring groups, so the prescribed-only median of 0 mm (groups placed flush or overlapping) leaves almost no approach margin, whereas the optimizer's added separation provides a more robust envelope. Clearance beyond the gripper's footprint adds little further benefit and can even lengthen robot travel between widely separated groups, so the relevant advantage is reaching sufficient rather than maximal separation.

7 Discussion

Our results show that allowing authors to delegate spatial detail to a downstream executor reduces task completion time and perceived workload, increases inter-group clearance in the end-to-end workflow, and is preferred by most users.

7.1 Fixture-Connection Speedup (RQ1, H1)

The dual-mode advantage was strongest for assembly tasks ($p = .002$), concentrated at fixture-connection levels (L2/L3 in T2 and T3, 2.5–2.7$\times$ faster). When two fixtures are assembled together, the connection physically repositions one fixture to align with the other, displacing the subassemblies arranged on it. In the prescribed-only condition, confirmation rechecks the displaced arrangement constraints and fails, forcing participants to regrab and readjust every subassembly back into tolerance. In dual-mode, those arrangement constraints are delegated and pass regardless of displacement, so confirmation succeeds immediately. Arrangement tasks showed a non-significant reduction ($p = .23$), likely because arrangement-level placement is inherently faster (fewer parts to coordinate) and the delegation benefit is proportionally smaller.

7.2 Delegation Scales with Complexity

Delegation increased with structure complexity (T1: 57.5%, T2: 80.8%, T3: 90.0%). Toggle counts followed the same pattern, confirming that delegation reflects active participant choices rather than default-to-delegated bias. We interpret this trend through the lens of cognitive load and levels of automation [38]: as part count, fixtures, and hierarchy depth grow, the bookkeeping required to prescribe every pose rises faster than the cost of delegating it, so authors shift authority to the executor wherever exact placement is not essential. This complements task-level findings that operators accept greater autonomy when it improves team efficiency rather than retained control [18], extending them from temporal task allocation to spatial pose delegation. This trend is an observation within our polycube assembly tasks: whether it reflects a general convergence toward delegation across structure variables and other task domains remains future work.

7.3 End-to-End Workflow Clearance (RQ2, H2)

H2 compares end-to-end workflows, not human placement with an optimizer in isolation. Prescribed-only authoring fixes every pose, whereas dual-mode authoring exposes delegated poses to optimization. The resulting clearance increase is therefore expected by construction; the contribution is an authoring model that separates relational intent from spatial detail so optimization can act.

7.4 Workload and Preference (RQ3/H3, RQ4/H4)

Workload was lower for dual-mode on all six NASA-TLX subscales, with five reaching significance (H3), consistent with prior work on the relationship between decision authority and operator workload [28]. Aoyama et al. [3] found visualization-dependent differences in physical demand, temporal demand, and effort, but not mental demand, performance, or frustration. Here, instruction mode affected the total score and five subscales. 85% of participants preferred dual-mode ($p = .003$), citing reduced spatial reasoning burden and the ability to focus on assembly relationships rather than exact poses. The three participants who preferred prescribed-only cited familiarity and simplicity, suggesting that the additional interface complexity of dual-mode carries its own cognitive cost. That the dual-mode time advantage grew with structure complexity (22% for T1 to 36% for T3) is consistent with this trade-off: for simpler structures, the cognitive overhead of managing delegation may offset its spatial planning benefit, while for complex structures, the reduction in spatial management dominates.

7.5 When Human-in-the-Loop Authoring Is Needed

Our work targets the middle of the automation spectrum [38, 13], between fully automated systems that derive constraints from CAD models [8, 63, 12, 4, 45] and manual waypoint programming. Following the function allocation principle [16], the author provides relational intent while the executor optimizes spatial details. This division is likely most valuable when no CAD model exists (maintenance, repair, rapid prototyping), when subassemblies share a workspace requiring reachability optimization, or when domain knowledge must be encoded by a human; we offer these as motivating scenarios rather than settings we evaluate here (Sec. 8). Delegated constraints are also portable: inter-part relationships remain valid across workspaces, reducing respecification to hierarchy adjustments rather than reauthoring every pose.

7.6 Constraint-Based XR Authoring Design Guidelines

Based on our findings, we propose the following design guidelines for XR constraint-authoring interfaces. Each is grounded in an XR-specific aspect of our system and, we argue, generalizes beyond assembly to spatial interfaces that must communicate nested structure and shared decision authority in 3D:

Support within-group delegation granularity. Participants prescribed fixtures while delegating children arranged on them (57.5–90.0% delegation rate), demonstrating selective, context-dependent delegation rather than all-or-nothing use. Interfaces that offer only global delegation modes would not support this strategy.

Build hierarchy through embodied bimanual actions. Authors constructed the constraint tree by holding one part in each hand and releasing the child to nest it (Sec. 3), an action distinct from physical snapping that declares semantic grouping without geometric contact. Mapping a binary parent–child relation onto two hands makes structure-building direct rather than menu-driven; XR interfaces that assemble nested or relational structures should consider bimanual selection for the same reason.

Separate hierarchy encoding from delegation encoding. Our pilot revealed that a single visual channel (nested bounding hulls) could not communicate both parent–child relationships and prescribed/delegated state (Sec. 4.1); we split the encoding into persistent animated lines for hierarchy plus on-demand bounding hulls for delegation. Constraint-based authoring interfaces should use distinct visual channels for structure (which parts belong together) and authority (who determines their pose).

Preserve hierarchy in downstream optimization. Constraint-based authoring produces hierarchical specifications that downstream optimizers must respect. Our optimizer resolves placements layer by layer (parents before children), and the 100% feasibility rate depends on this ordering. A flat optimizer that ignores hierarchy depth would place children without knowing their parent surface, producing infeasible configurations.

8 Limitations and Future Work

8.1 Study Limitations

All trials used polycube assets with a single simulated UR5 robot and gripper, with at most 15 parts and five hierarchy levels. More complex tasks, diverse parts and robots could test generalizability, possibly necessitating support for filtering and searching. Both conditions share the same hierarchy visualization, but dual-mode includes delegated bounding hulls. This additional visual feedback is a potential confound: dual-mode provides richer spatial information (on-demand bounding hulls) that could aid spatial reasoning. However, the primary H1 effect is concentrated at fixture-connection assembly levels where assembling fixtures together displaces subassemblies (delegated arrangement constraints absorb the displacement, while prescribed constraints require manual repair), not spatial reasoning aided by hulls. One participant (P20) never delegated in dual-mode yet still experienced the dual-mode interface; P20's data provides a natural control for the visual feedback confound and is retained under intent-to-treat.

As discussed in Sec. 4.2, we did not include a delegation-only condition: assembly connections leave no placement freedom to delegate, and dual-mode already allowed participants to delegate every delegable group. Future work could test whether visual feedback alone (bounding hulls independent of delegation) accounts for part of the observed effect at arrangement levels. Structure (T1–T3) confounds parts, levels, and fixtures, inherent to the domain; we treat it as an ordinal complexity factor.

8.2 Planner Evaluation Scope

Our evaluation assesses pose feasibility in simulation, not grasp synthesis or post-assembly manipulation. The MuJoCo executor confirms that authored specifications yield collision-free, kinematically feasible pick-and-place, but it does not model the sensing and actuation noise of a physical robot. We plan on replaying the same specifications on robot hardware to test whether poses certified in simulation survive calibration error and gripper compliance. We also plan on extending the optimizer to tight-tolerance domains and to multi-robot work allocation, which assigns delegated groups across manipulators by reachability.

8.3 Future Directions

Our constraint model couples semantic grouping with kinematic parenting: nesting a part under a group both declares a spatial relationship and moves the child with the parent. This coupling creates friction when the author wants to relate two parts without either serving as the other's fixed reference; participants worked around it by parenting purely for organization, nesting parts they kept far apart (Sec. 5.6). Future versions could decouple the constraint graph from the kinematic hierarchy, constraining related parts symmetrically. Delegation could become more expressive: authors could constrain a subset of a group's degrees of freedom via per-axis position and orientation masks, sculpt delegated regions top-down (controlling the shape of displayed hulls), or assign a delegated placement to any of a set of fixtures.

9 Conclusions

We presented an XR constraint-authoring interface that separates the author who specifies constraints from the executor who carries them out. The author builds a hierarchical constraint tree through bimanual nesting, indicating per group whether to prescribe or delegate its 6DoF pose relative to a parent, thus retaining or giving up specification authority. A within-subjects study showed that dual-mode authoring significantly reduced task completion time and perceived workload and was preferred by most participants.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Award CMMI-2037101.

References

[1] L. L. Ames and E. J. George. Revision and verification of a seven-point workload estimate scale. *Technical Information Manual, Air Force Flight Test Center, Edwards Air Force Base*, 1993.

[2] J. Ao, F. Wu, Y. Wu, A. Swiki, and S. Haddadin. LLM-as-BT-Planner: Leveraging LLMs for behavior tree generation in robot task planning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1233–1239, 2025. doi: 10.1109/ICRA55743.2025.11128454

[3] S. Aoyama, J.-S. Liu, P. Wang, S. Jain, X. Wang, J. Xu, S. Song, B. Tversky, and S. K. Feiner. Asynchronously assigning, monitoring, and managing assembly goals in virtual reality for high-level robot teleoperation. In *Proc. IEEE VR*, pp. 450–460. IEEE, 2024. doi: 10.1109/VR58804.2024.00066

[4] S. Asif, M. Bueno, P. Ferreira, P. Anandan, Z. Zhang, Y. Yao, G. Ragunathan, L. Tinkler, M. Sotoodeh-Bahraini, N. Lohse, P. Webb, W. Hutabarat, and A. Tiwari. Rapid and automated configuration of robot manufacturing cells. *Robotics and Computer-Integrated Manufacturing*, 92:102862, 2025. doi: 10.1016/j.rcim.2024.102862

[5] S. V. Babu, W.-A. Hsieh, and J.-H. Chuang. The benefits of near-field manipulation and viewing to distant object manipulation in VR. In *Proc. IEEE VR*. IEEE, 2024. doi: 10.1109/VR58804.2024.00062

[6] D. Bauer, C. Zheng, O.-H. Kwon, and K.-L. Ma. A multi-layout design for immersive visualization of hierarchical network data. In *Proc. IEEE ISMAR*. IEEE, 2024. doi: 10.1109/ISMAR62088.2024.00120

[7] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982. doi: 10.1145/357306.357310

[8] K. Brown, D. M. Asmar, M. Schwager, and M. J. Kochenderfer. Large-scale multi-robot assembly planning for autonomous manufacturing. *Robotics and Autonomous Systems*, 194:105179, 2025. doi: 10.1016/j.robot.2025.105179

[9] M. Brysbaert and M. Stevens. Power analysis and effect size in mixed effects models: A tutorial. *Journal of Cognition*, 1(1):9, 2018. doi: 10.5334/joc.10

[10] W. Büschel, S. Vogt, and R. Dachselt. Augmented reality graph visualizations. *IEEE Computer Graphics and Applications*, 39(3):29–40, 2019. doi: 10.1109/MCG.2019.2897927

[11] Y. Cao, T. Wang, X. Qian, P. S. Rao, M. Wadhawan, K. Huo, and K. Ramani. GhostAR: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality. In *Proc. ACM UIST*. ACM, New York, 2019. doi: 10.1145/3332165.3347902

[12] C. Cella, M. B. Robin, M. Faroni, A. M. Zanchettin, and P. Rocco. Digital model-driven genetic algorithm for optimizing layout and task allocation in human-robot collaborative assemblies. In *Proc. IEEE ICRA*, 2025. doi: 10.1109/ICRA55743.2025.11127401

[13] M. Chiou, N. Hawes, and R. Stolkin. Mixed-initiative variable autonomy for remotely operated mobile robots. *ACM Transactions on Human-Robot Interaction*, 10(4), 2021. doi: 10.1145/3472206

[14] D. Coleman, I. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: A MoveIt! case study. *Journal of Software Engineering for Robotics*, 5(1):3–16, 2014. doi: 10.6092/JOSER_2014_05_01_P3

[15] M. Fechter, B. Schleich, and S. Wartzack. Comparative evaluation of WIMP and immersive natural finger interaction: A user study on CAD assembly modeling. *Virtual Reality*, 25:1263–1277, 2021. doi: 10.1007/s10055-021-00543-0

[16] P. M. Fitts. Human engineering for an effective air-navigation and traffic-control system. Technical report, National Research Council, Washington, DC, 1951.

[17] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005. doi: 10.1137/S0036144504446096

[18] M. C. Gombolay, R. A. Gutierrez, S. G. Clarke, G. F. Sturla, and J. A. Shah. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots*, 39(3):293–312, 2015. doi: 10.1007/s10514-015-9457-9

[19] M. Görner, R. Haschke, H. Ritter, and J. Zhang. MoveIt! task constructor for task-level motion planning. In *Proc. IEEE ICRA*, pp. 190–196. IEEE, 2019. doi: 10.1109/ICRA.2019.8793898

[20] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4):486–517, 1987. doi: 10.1080/00222895.1987.10735426

[21] M. Hagenow, E. Senft, R. Radwin, M. Gleicher, M. Zinn, and B. Mutlu. A system for human-robot teaming through end-user programming and shared autonomy. In *Proc. ACM/IEEE HRI*. ACM, New York, 2024. doi: 10.1145/3610977.3634965

[22] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, eds., *Human Mental Workload*, pp. 139–183. North-Holland, Amsterdam, 1988.

[23] L. S. Homem de Mello and A. C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. Robotics and Automation*, 7(2):228–240, 1991. doi: 10.1109/70.75905

[24] B. Ikeda, M. Gramopadhye, L. Nekervis, and D. Szafir. MARCER: Multimodal augmented reality for composing and executing robot tasks. In *Proc. ACM/IEEE HRI*. IEEE, 2025. doi: 10.1109/HRI61500.2025.10974232

[25] B. Ikeda and D. Szafir. PRogramAR: Augmented reality end-user robot programming. *ACM Trans. Hum. Robot Interact.*, 2024. doi: 10.1145/3640008

[26] L. Johannsmeier and S. Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48, 2017. doi: 10.1109/LRA.2016.2535907

[27] L. Joos, S. Jaeger-Honz, F. Schreiber, D. Keim, and K. Klein. Visual comparison of networks in VR. *IEEE Trans. Visualization and Computer Graphics*, 2022. doi: 10.1109/TVCG.2022.3203001

[28] M. Karakikes and D. Nathanael. The effect of cognitive workload on decision authority assignment in human-robot collaboration. *Cognition, Technology & Work*, 2022. doi: 10.1007/s10111-022-00719-x

[29] M. Krichenbauer, G. Yamamoto, T. Taketomi, C. Sandor, and H. Kato. Augmented reality versus virtual reality for 3D object manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1038–1048, 2018. doi: 10.1109/TVCG.2017.2658570

[30] J. Krieglstein, G. Held, B. A. Bálint, F. Nägele, and W. Kraus. Skill-based robot programming in mixed reality with ad-hoc validation using a force-enabled digital twin. In *Proc. IEEE ICRA*. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10161095

[31] C.-Y. Lee, W.-A. Hsieh, D. Brickler, S. V. Babu, and J.-H. Chuang. Design and empirical evaluation of a novel near-field interaction metaphor on distant object manipulation in VR. In *Proc. ACM SUI*, pp. 1–12. ACM, New York, 2021. doi: 10.1145/3485279.3485296

[32] S.-H. Lee, T. Jin, J. H. Lee, and S.-H. Bae. WireSketch: Bimanual interactions for 3D curve networks in VR. In *Proc. ACM UIST*. ACM, New York, 2022. doi: 10.1145/3526114.3558726

[33] Y. Li, S. Agrawal, J.-S. Liu, S. K. Feiner, and S. Song. Scene editing as teleoperation: A case study in 6DoF kit assembly. In *Proc. IEEE IROS*. IEEE, 2022. doi: 10.1109/IROS47612.2022.9982158

[34] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 163–169, 1987. doi: 10.1145/37401.37422

[35] R. Lunding, S. Hubenschmid, T. Feuchtner, and K. Grønbæk. ARTHUR: Authoring human-robot collaboration processes with augmented reality using hybrid user interfaces. *Virtual Reality*, 29(2):73, 2025. doi: 10.1007/s10055-025-01149-6

[36] M. N. Lystbæk, T. Mikkelsen, R. Krisztandl, E. J. Gonzalez, M. Gonzalez-Franco, H. Gellersen, and K. Pfeuffer. Hands-on, hands-off: Gaze-assisted bimanual 3D interaction. In *Proc. UIST*. ACM, New York, 2024. doi: 10.1145/3654777.3676331

[37] MeasuringU. *10 Things to Know about the NASA TLX*. https://measuringu.com/nasa-tlx/, last accessed on 03/15/2026.

[38] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 30(3):286–297, 2000. doi: 10.1109/3468.844354

[39] M. Peters, B. Laeng, K. Latham, M. Jackson, R. Zaiyouna, and C. Richardson. A redrawn vandenberg and kuse mental rotations test - different versions and factors that affect performance. *Brain and Cognition*, 28(1):39–58, 1995. doi: 10.1006/brcg.1995.1032

[40] J. S. Pierce, B. C. Stearns, and R. Pausch. Voodoo dolls: Seamless interaction at multiple scales in virtual environments. In *Proc. SI3D*, pp. 141–145. ACM, New York, 1999. doi: 10.1145/300523.300540

[41] H. Pohl, K. Lilija, J. McIntosh, and K. Hornbæk. Poros: Configurable proxies for distant interactions in VR. In *Proc. ACM CHI*. ACM, New York, 2021. doi: 10.1145/3411764.3445685

[42] D. Porfirio, M. Roberts, and L. M. Hiatt. Goal-oriented end-user programming of robots. In *Proc. ACM/IEEE HRI*. ACM, New York, 2024. doi: 10.1145/3610977.3634974

[43] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proc. UIST*, pp. 79–80. ACM, New York, 1996. doi: 10.1145/237091.237102

[44] I. Quilez. Smooth minimum. https://iquilezles.org/articles/smin/, 2013. Accessed: 2026-03-11.

[45] Realtime Robotics. Resolver: Automated motion planning and optimization, 2025. Accessed: 2026-02-12.

[46] P. Sasikumar, R. Hajika, K. Gupta, T. S. Gunasekaran, Y. S. Pai, H. Bai, S. Nanayakkara, and M. Billinghurst. A user study on sharing physiological cues in VR assembly tasks. In *Proc. IEEE VR*. IEEE, 2024. doi: 10.1109/VR58804.2024.00096

[47] A. Schoen, C. Henrichs, M. Strohkirch, and B. Mutlu. Authr: A task authoring environment for human-robot teams. In *Proc. ACM UIST*. ACM, New York, 2020. doi: 10.1145/3379337.3415872

[48] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons, and S. Singh. Coordinated multiagent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE*, 94(7):1425–1444, 2006. doi: 10.1109/JPROC.2006.876966

[49] E. Senft, M. Hagenow, R. Radwin, M. Zinn, M. Gleicher, and B. Mutlu. Situated live programming for human-robot collaboration. In *Proc. ACM UIST*. ACM, New York, 2021. doi: 10.1145/3472749.3474773

[50] E. Senft, M. Hagenow, K. Welsh, R. Radwin, M. Zinn, M. Gleicher, and B. Mutlu. Task-level authoring for remote robot teleoperation. *Frontiers in Robotics and AI*, 8, 2021. doi: 10.3389/frobt.2021.707149

[51] Stereo Optical Co. Inc. *Original Stereo Fly Stereotest*. https://www.stereooptical.com/products/stereotests-color-tests/original-stereo-fly/, last accessed on 03/15/2026.

[52] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. CHI*, pp. 265–272. ACM, New York, 1995. doi: 10.1145/223904.223938

[53] R. Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics. https://drake.mit.edu, 2019.

[54] Y. Tian, K. D. D. Willis, B. Al Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda, H. Li, S. Chitta, and W. Matusik. ASAP: Automated sequence planning for complex robotic assembly with physical feasibility. In *Proc. IEEE ICRA*. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611595

[55] C. Tie, S. Sun, Y. Lin, Y. Wang, Z. Li, Z. Zhong, J. Zhu, Y. Pang, H. Chen, J. Chen, R. Wu, and L. Shao. Manual2Skill++: Connector-aware general robotic assembly from instruction manuals via vision-language models. *arXiv preprint arXiv:2510.16344*, 2025. doi: 10.48550/arXiv.2510.16344

[56] C. Tie, S. Sun, J. Zhu, Y. Liu, J. Guo, Y. Hu, H. Chen, J. Chen, R. Wu, and L. Shao. Manual2Skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models. In *Proc. RSS*, 2026. doi: 10.48550/arXiv.2502.10090

[57] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *Proc. IEEE IROS*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109

[58] N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *Proc. IEEE ICRA*, pp. 1970–1975. IEEE, 2013. doi: 10.1109/ICRA.2013.6630839

[59] V. Vuibert, W. Stuerzlinger, and J. R. Cooperstock. Evaluation of docking task performance using mid-air interaction techniques. In *Proc. ACM SUI*, pp. 44–52. ACM, 2015. doi: 10.1145/2788940.2788950

[60] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi: 10.1007/s10107-004-0559-y

[61] L. W. Wang and P. Abtahi. GhostObjects: Instructing robots by manipulating spatially aligned virtual twins in augmented reality. In *Proc. ACM UIST Adjunct*. ACM, 2025. doi: 10.1145/3746058.3758451

[62] P. Wang, S. Jain, M. Li, S. Aoyama, X. Wang, S. Song, J.-S. Liu, and S. K. Feiner. Built to order: A virtual reality interface for assigning high-level assembly goals to remote robots. In *Proc. ACM SUI*. ACM, New York, 2023. doi: 10.1145/3607822.3618005

[63] J. Wessén, M. Carlsson, C. Schulte, P. Flener, F. Pecora, and M. Matskin. A constraint programming model for the scheduling and workspace layout design of a dual-arm multi-tool assembly robot. *Constraints*, 28(2):71–104, 2023. doi: 10.1007/s10601-023-09345-4

[64] S. Xu, S. Moore, and A. Cosgun. Shared-control robotic manipulation in virtual reality. In *Proc. HORA*, 2022. doi: 10.1109/HORA55278.2022.9800046

[65] B. Yang, X. He, C. Zou, J.-S. Liu, B. Tversky, and S. Feiner. Virtual reality task guidance through relative 6DoF pose specification. In *2025 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 901–902, 2025. doi: 10.1109/ISMAR-Adjunct68609.2025.00247

[66] B. Yang, X. He, C. Zou, J.-S. Liu, B. Tversky, and S. K. Feiner. Bimanual XR specification of relative and absolute assembly hierarchies for teleoperation. In *Proc. IEEE Conf. Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 1033–1034. IEEE, 2026.

[67] B. Yang, J.-S. Liu, X. He, B. Tversky, and S. Feiner. Permanent proxies for bimanual selection and manipulation in VR. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 1330–1331, 2025. doi: 10.1109/VRW66409.2025.00311

[68] F. Zhang, K. Katsuragawa, and E. Lank. Conductor: Intersection-based bimanual pointing in augmented and virtual reality. *Proc. ACM Hum. Comput. Interact.*, 6(ISS), 2022. doi: 10.1145/3567713

[69] Z. Zhao, S. Cheng, Y. Ding, Z. Zhou, S. Zhang, D. Xu, and Y. Zhao. A survey of optimization-based task and motion planning: From classical to learning approaches. *IEEE/ASME Transactions on Mechatronics*, 2024. doi: 10.1109/TMECH.2024.3452509