Delegate the Details: Flexible Assembly Constraint Authoring for Collaborative XR

Category: Research

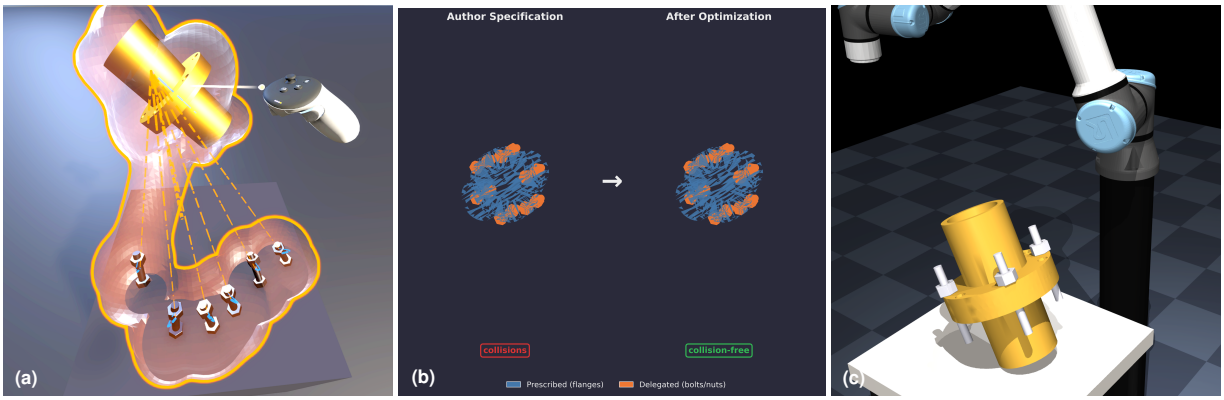


Figure 1: End-to-end pipeline. (a) An author manipulates a delegated group (orange volume) in XR; animated lines show parent–child relationships. (b) The workspace optimizer resolves delegated poses for collision avoidance, support, and reachability. (c) A simulated UR5 executes the planned assembly in MuJoCo.

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. Both modes can be mixed within a single task. We conducted a within-subjects study comparing prescribed-only authoring against dual-mode authoring. Results show that dual-mode authoring significantly reduced authoring task completion time and perceived workload. Most participants significantly preferred dual-mode, with delegation rate scaling with the complexity of authored structures. We also evaluated both conditions offline with a robot executor, feeding authored constraints through a workspace optimizer and motion planner. Delegated constraints achieved significantly higher inter-group clearance than prescribed-only specifications.

Index Terms: Extended reality collaboration, Human–Robot Interaction, assembly constraints, task specification, constraint authoring.

1 Introduction

Extended reality (XR) interfaces for assembly task specification have evolved from low-level direct control to high-level goal specification [3, 58, 57]. 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 [48].

However, current goal-specification interfaces typically require users to define the 6DoF pose of every part or operation [31, 3, 28]. While straightforward for a single part, this approach becomes cumbersome for multi-part assembly tasks involving hierarchical structures [62]. Consider a manufacturing assembly line where components are first assembled into subassemblies on dedicated fixtures, then the loaded fixtures are combined into a final product: the user must specify each part's exact pose even though the essential constraints are which parts connect and how subassemblies relate to each other. Many of these 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 control over part pose to delegation of determining part pose to the 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 interface organizes assembly elements into a hierarchy of *groups*, where each group defines a reference frame and can contain any number of child groups (Fig. 2). Each group corresponds to a single part that defines its reference frame. Each group carries a constraint mode: in *prescribed* mode, the author specifies the 6DoF pose of the group relative to its parent; in *delegated* mode, the author entrusts the executor to determine the pose. A within-group toggle switches between prescribed and delegated modes, and both modes can be mixed within a single task (see Fig. 1b).

Authors build the hierarchy through *bimanual nesting* (Sec. 3), where simultaneously holding one part with each controller establishes a semantic parent–child relationship between them (Fig. 1a). Two complementary visualizations communicate the resulting constraint structure: *animated directional lines* persistently encode the hierarchy, while *enclosing volumes* appear on demand around del-

egated groups to mark that the executor will determine the final configuration. Delegated constraints give the planner additional degrees of freedom, expanding the feasible solution space and reducing planning failures for complex assemblies.

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 constraint modes that can be mixed within a single assembly.
- A within-subjects study ($n = 20$) comparing prescribed-only authoring against dual-mode authoring across three hierarchical structures in order to test whether dual-mode authoring reduces task completion time and perceived workload.
- Analysis of how structural complexity affects efficiency across assembly and arrangement task types.
- An offline planner evaluation assessing whether delegated constraints improve downstream robot planning feasibility.

2 Related Work

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

2.1 Constraint Specification for Collaborative Assembly

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 [65], with recent work scaling to hundreds of robots [8]. Constraint programming jointly optimizes workspace layout and task scheduling [59, 12], and automated cell configuration systems derive robot placements from product models [4]. Vision–Language models (VLMs) extract specifications from instruction manuals [53, 52], and approaches based on Large Language Models (LLMs) generate assembly plans from task descriptions [2]. These systems assume the assembly goal is known and fixed: the system *extracts* a specification rather than *creating* one.

Full human control. At the other end, goal-based teleoperation interfaces allow users to specify desired end states rather than trajectories [58, 31]. Aoyama et al. [3] extended this to asynchronous goal assignment where users specify 6DoF assembly targets in XR, supporting concurrent specification and execution. End-user authoring environments enable non-experts to program workflows through workflow authoring [45], embodied XR demonstration [11], situated programming [47], multimodal XR with LLM-based composition [23, 22], XR skill-parameter visualization with digital-twin validation [28], and hybrid XR authoring for HRI [33]. XR assembly interfaces have been evaluated for CAD modeling [15] and collaborative tasks [44].

Mixed-initiative approaches. Between these extremes, shared-control approaches blend user input with autonomous capabilities [46, 60]. Senft et al. [48] introduced task-level authoring for variable robot autonomy, Hagenow et al. [20] combined end-user programming with shared autonomy, and Porfirio et al. [40] demonstrated goal-oriented programming at variable granularity. Hierarchical planning frameworks decompose collaborative assembly into task allocation between human and robot [24].

Yang et al. [61, 62] introduced within-group specification modes for hierarchical assembly in XR, allowing users to 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"). However, that work lacked a user study. 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. Our work extends this by introducing constraint-based goal specification: rather than demonstrating a single desired configuration, users specify which spatial relationships must be preserved while delegating remaining decisions to the executor.

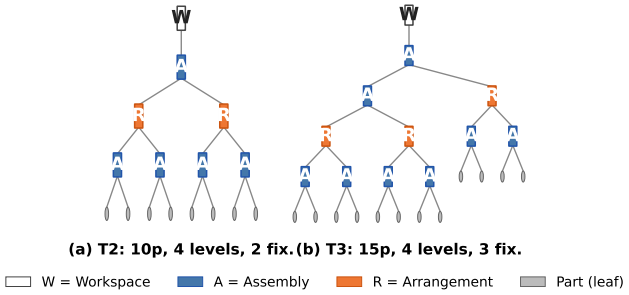


Figure 2: Task hierarchies. T1 (not shown) is flat: three pair assemblies on the table. (a) T2: 10 parts, 2 fixtures, 4 levels. (b) T3: 15 parts, 3 fixtures, 4 levels. Blue = assembly; orange = arrangement.

XR interaction techniques. Our interface uses bimanual nesting to build the constraint hierarchy. Bimanual interaction in XR leverages natural two-handed coordination [19] through techniques spanning proxies [38, 39], miniature workspaces [50], extended reach [41], and persistent spatial references [38]. Recent work has combined bimanual input with pointing [64], gaze [34], and lines [30]. Object manipulation spans direct grab [27], near-field [29], 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.

2.2 Assembly Feasibility Evaluation

Assembly feasibility evaluation 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 [51] generates physically feasible assembly sequences by ensuring intermediate physical stability and realizability. Inverse reachability maps [55] 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 final configuration, thus expanding the feasible solution space. We adopt established feasibility evaluation methods [51, 55] 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 enclosing volumes that are shown on demand, and exports specifications for downstream execution.

3.1 Hierarchical Constraint Model

An assembly is represented as a tree of *groups* (Fig. 2). Each group corresponds to a single part that defines its reference frame, and may have any number of child groups nested beneath it; child groups express their poses relative to their parent's reference frame. The root group represents the workspace (e.g., the assembly table).

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 must reproduce this exact configuration.
- **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 is within-group: it describes this 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 assembly: for example, two subassemblies on a shared fixture might have different delegation states while the fixture itself is prescribed.

3.2 Interaction Design

Authors build the constraint hierarchy through the following mechanisms (Fig. 3):

Bimanual nesting. To establish parent–child relationships (e.g., grouping subassemblies under a shared parent), the author holds one part with each controller and releases the intended child. The released part's group is reparented under the held part's group. This two-handed gesture creates a semantic parent–child relationship distinct from physical snapping: it defines which subassemblies belong together in the constraint tree without requiring geometric contact. Un-nesting is performed by holding the child and the workspace root, then releasing the child.

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 are recorded as a compound undo entry, so a single undo press reverts both.

Undoing nesting. Pressing either thumbstick reverts the most recent nesting operation from a global stack. For cyclic auto-fixes, 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 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 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 is rejected if parts collide or lack support, with visual and haptic feedback indicating the failure reason.

3.3 Visualization

The interface provides two complementary visual encodings that communicate the constraint hierarchy and delegation state (see Fig. 1):

Animated directional lines connect each child group to its parent using animated chevrons that flow parent-to-child, color-coded blue (prescribed) or orange (delegated). We chose animated directional edges following Büschel et al.'s [10] finding that animated edges were effective for directed graph perception in XR, and Joos et al.'s [25] finding that node-link representations outperform matrix encodings for network comparison tasks in XR. Lines are always visible, providing a persistent hierarchy overview; no lines are drawn to the workspace root.

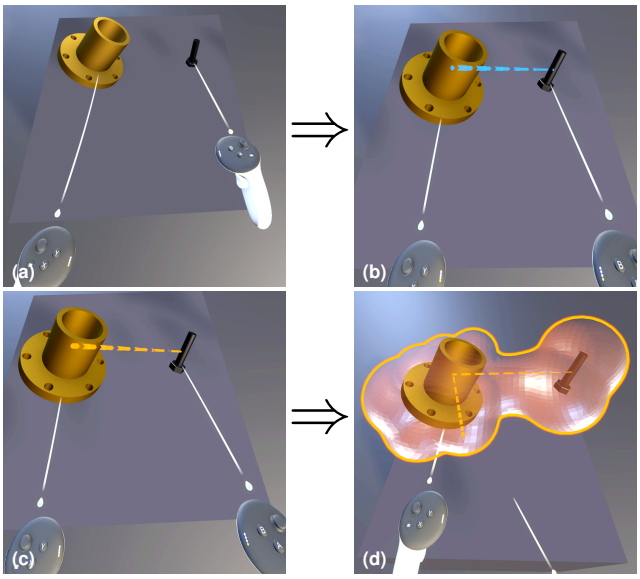


Figure 3: Bimanual nesting and mode toggling. (a) Author holds two parts, one per hand. (b) Releasing the right-hand part nests it under the left, creating a prescribed (blue) parent–child link. (c) B/Y toggles the child to delegated (orange). (d) An enclosing volume appears, marking the group for executor-determined pose.

Demand-driven enclosing volumes appear on delegated groups when either controller is within 12 cm of the group's centroid (and hidden during grab), marking where the executor determines final configuration. Each volume is an implicit isosurface [7]: per-part voxel centers contribute sphere signed distance field (SDF) primitives blended via polynomial smooth-min [42], with the mesh extracted by GPU marching cubes [32]. The resulting surface conforms organically to the group's geometry rather than enclosing it in a bounding box, providing a tighter visual proxy of group extent. This hybrid avoids clutter by combining persistent lines for hierarchy overview with on-demand enclosures for group membership, following evidence that multi-layout designs outperform single representations for hierarchical visualization in immersive environments [6]. Prescribed groups are never shown as volumes, so volumes are a visual marker of delegation.

Confirmation feedback uses per-part outline states: green (confirmed), gold (changed since last confirmation), and red (colliding).

3.4 End-to-End Pipeline

The interface connects to an end-to-end pipeline from XR constraint authoring to simulated execution.

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 determines feasible poses satisfying collision, support, and reachability constraints using a hybrid solver cascade. The pose of each delegated group serves as a decision variable; prescribed poses pass through directly.

Motion Planning: Optimized poses are translated into MoveIt [14] planning goals using RRT-Connect (OMPL, 2s timeout), sequenced by Task Constructor [18].

Simulation: Trajectories are executed on a simulated Universal Robots UR5 arm in MuJoCo [54] to verify end-to-end feasibility.

4 User Study

4.1 Pilot User 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. Pilot participants completed the full study protocol and provided verbal feedback throughout.

Visualization. The initial interface used nested hull enclosures (semi-transparent bounding volumes) to communicate hierarchy and delegation state; 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 enclosing volumes appearing only on delegated groups when the controller is nearby (Sec. 3).

Interaction. Pilot sessions revealed that participants frequently attempted proximity-based nesting rather than the bimanual trigger gesture. We refined the visual and haptic feedback to make the bimanual nesting gesture 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 User Study

We conducted a within-subjects study, approved by our institutional reivew board, comparing prescribed-only authoring against dual-mode (prescribed and delegated) authoring for hierarchical assembly constraint specification. To isolate the effect of delegation, the study adapts the full interface described above in several ways:

Assembly mechanics. All study trials use *polycube pieces* to control for part geometry, with connection points defined in trial configuration files: pentacubes (5 unit cubes) for regular parts and decacubes (10 unit cubes) for fixtures, providing uniform geometric complexity with full 6DoF relevance.

Parts snapped together when brought within ± 2 cm translational and $\pm 10^\circ$ rotational tolerance of defined connection points (informed by mid-air docking thresholds reported by Vuibert et al. [56], 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.

Task guidance. Assembly guidance was provided through floating World-in-Miniature (WiM) displays [50]: each task received its own WiM showing the target configuration at reduced scale. WiMs were revealed progressively as dependency-graph prerequisites were satisfied, allowing independent branches to progress at different rates. WiMs were placed along the near edge pf a virtual table, building progressively as tasks unlocked and consolidating as merges occurred.

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), ensuring identical runtime behavior across conditions.

Conditions. Both conditions share all adaptations above, the same hierarchy visualization (animated directional lines), bimanual nesting, and confirmation mechanics. Both conditions also share identical visual feedback for confirmed (green), changed (gold), and colliding (red) states. In the *prescribed-only* condition, the delegation toggle is locked (all groups remain prescribed, no delegated volumes appear); participants specify all poses themselves. In the *dual-mode* condition, all groups begin as prescribed; the toggle is unlocked with a button press and delegated volumes provide additional visual feedback; participants can mark groups as delegated at arrangement levels, signaling that the executor should determine the pose. The independent variable is therefore isolated to delegation choice. Block order was counterbalanced (10 per ordering), and each block was preceded by condition-specific training (three practice trials of increasing complexity).

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). This is where delegation and the optimizer operate in dual-mode.

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 dual-mode workflow (authoring + optimization) produce better downstream robot planning outcomes than prescribed-only authoring?
- **RQ3 (Workload):** Does dual-mode authoring reduce perceived workload?
- **RQ4 (Preference):** Do authors prefer dual-mode over prescribed-only authoring?

Confirmatory hypotheses (each at $\alpha = .05$; no cross-family correction):

- **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 condition $\times$ task_type interaction is reported as an exploratory contrast.
- **H2 (Robot Outcomes):** *The dual-mode workflow (authoring + optimization) produces superior end-to-end planning metrics compared to prescribed-only authoring* (inter-group clearance; Wilcoxon signed-rank on paired per-participant means).
- **H3 (Workload):** Unweighted NASA-TLX total score is lower in dual-mode than prescribed-only (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: Condition $\times$ task_type interaction (whether the dual-mode advantage differs between assembly and arrangement), delegation timing (early vs. late), hierarchy churn, per-structure breakdowns, MRT-A as moderator, 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 (resting on a surface, whether a fixture or the table; see 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) as it involves no meaningful spatial decision.

We designed three structures of increasing complexity, spanning two and four hierarchy levels (Fig. 2):

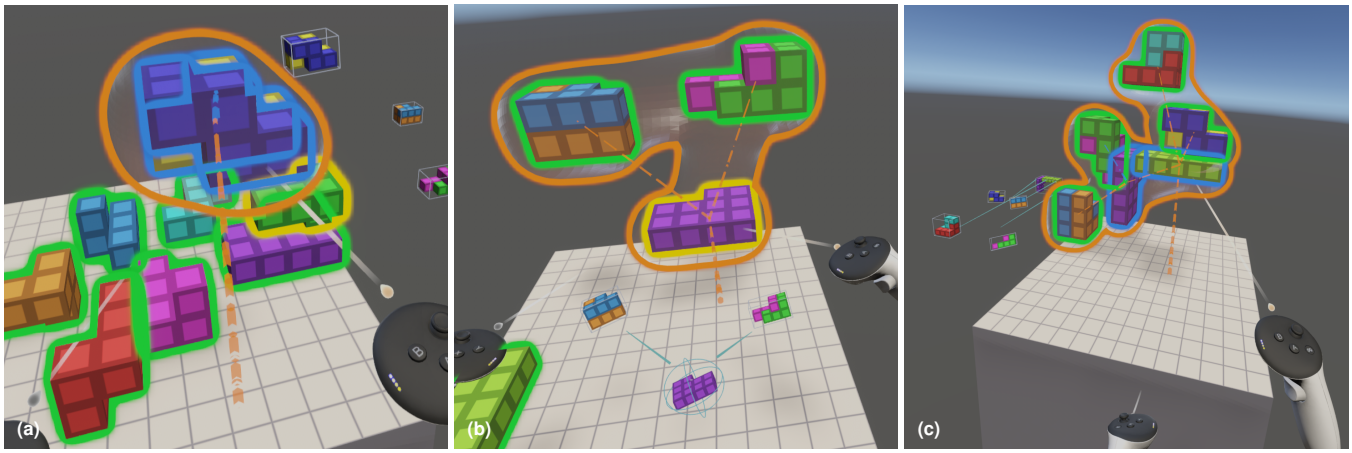


Figure 4: Progression of a T2 dual-mode trial. (a) Subassemblies assembled onto fixtures; orange volumes mark delegated groups, animated lines show hierarchy. (b) Fixtures lifted with subassemblies; blue lines indicate prescribed parts, unassembled parts remain on the table. (c) Fixtures connected into a single delegated structure (orange volume); guidance miniatures (left) show remaining tasks.

- **T1** (6 parts, 2 levels, 0 fixtures): Three pair assemblies arranged on the table. 3 assembly + 1 arrangement = 4 level-tasks.
- **T2** (10 parts, 4 levels, 2 fixtures): Assemblies arranged on two fixtures, fixtures merged, then the merged unit arranged on the table. 5 assembly + 3 arrangement = 8 level-tasks.
- **T3** (15 parts, 4 levels, 3 fixtures): Assemblies on three fixtures with multi-level merges. 8 assembly + 4 arrangement = 12 level-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 be disentangled. Per-structure breakdowns are reported for generalizability.

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 that contaminates 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 miniature reference.

There are 5 trials per block × 2 blocks = 10 scored trials per participant. 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), yielding a final sample of $n = 20$ (9 female, 11 male; ages 19–33, $M = 23.6$, $SD = 3.1$). Self-reported XR experience ranged from none (2) to daily or more (7), with the majority reporting monthly use (9) and two reported weekly use. All participants had normal or corrected-to-normal vision, passed Ishihara color-vision–deficiency screening, and completed the Stereo Fly test [49] and Mental Rotation Test (MRT-A) [37] as pre-questionnaires. Exclusion criteria included: cybersickness requiring session termination and failure to complete either practice block. Participants received modest monetary compenation (currency suppressed for blind review); 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 [35], 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 the second block. Block order was counterbalanced (10 per ordering); trial order within each block was semi-randomized (T3 not first).

4.7 Measures

Task completion time (primary dependent variable; H1): Time from when a level becomes active (all prerequisites in the depen-

dency 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 (H3): Six subscales adapted to 7-point Likert scales (1–7) [21], administered once per block.

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

Planner metrics (H2): Inter-group clearance, evaluated offline (Sec. 6).

Additional measures: grab count per arrangement task, dominant-hand path length from 30 Hz tracking, delegation rate and toggle count (dual-mode only), MRT-A score, and semi-structured interview feedback.

5 Results

5.1 Analysis Plan

Each confirmatory hypothesis tests a different construct and is analyzed independently at $\alpha = .05$; within H2, Holm–Bonferroni correction is applied across its dependent variables. The **H1 model** is a Gamma GLMM with log link, which directly models positive-valued 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) 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 in favor of 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), $z = -3.12$, $p = .002$. 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 condition $\times$ 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$\times$. In these levels, assembling two fixtures together physically repositions one fixture to align with the other, displacing the sub-assemblies arranged on them. In prescribed-only, confirmation rechecks the displaced arrangement constraints and fails, forcing participants to readjust every sub-assembly back into tolerance. In dual-mode, those arrangement constraints are delegated and pass regardless of displacement, so confirmation succeeds immediately. 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 $W = 192$, $p = .001$). 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 six favored dual-mode, confirming that delegation reduced perceived workload (Fig. 6a,b).

H4 (Preference). 17 of 20 participants (85%) preferred dual-mode (binomial $p = .003$; 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 $H = 42.3$, $p < .0001$). 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. Toggle counts confirm that delegation reflects active participant choices rather than passive acceptance of defaults.

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, though this interpretation requires further investigation.

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 sub-assemblies 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, $p = .004$). Block order was not significant ($p = .24$) and did not interact with condition. All confirmatory results (H1–H4) are robust across these specifications.

5.6 Qualitative Observations

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

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

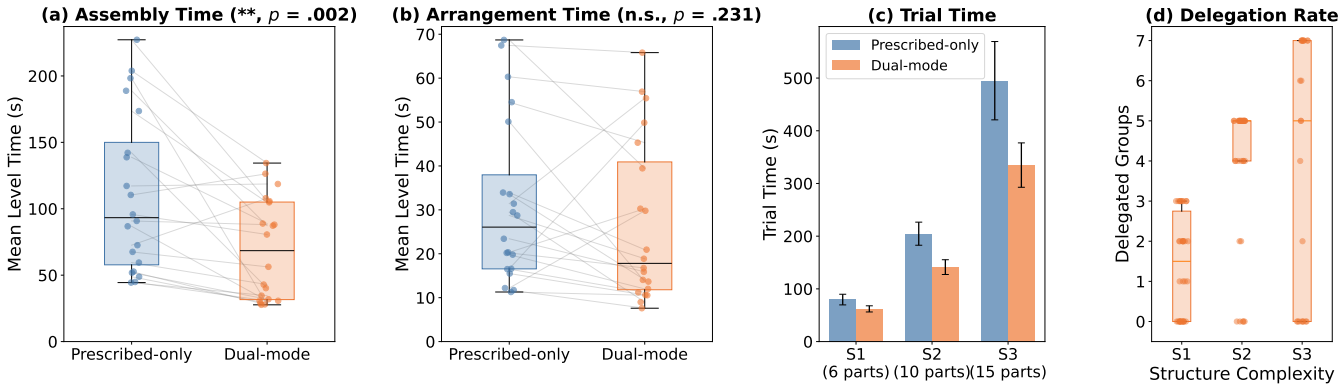


Figure 5: Task time and delegation. (a) Assembly time by condition (paired lines per participant; $p = .002$). (b) Arrangement time ($p = .23$, n.s.). (c) Trial time by structure complexity (± 1 SE). (d) Delegated groups per trial ($H = 42.3$, $p < .0001$).

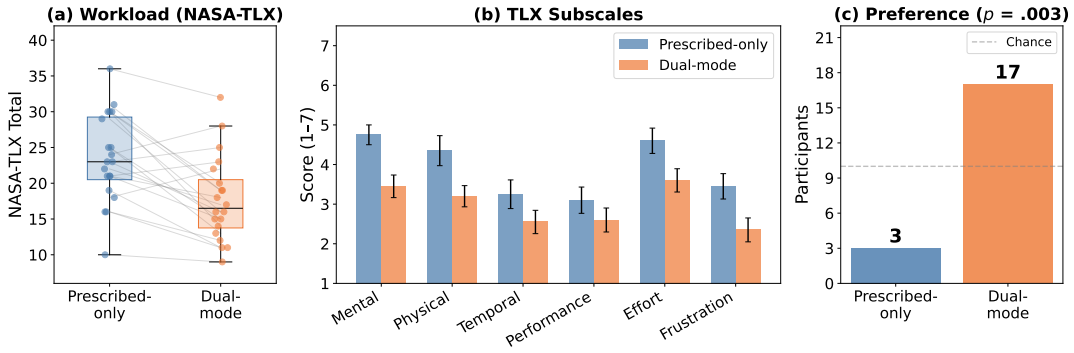


Figure 6: Workload and preference. (a) NASA-TLX totals by condition (paired lines). (b) TLX subscale means (± 1 SE). (c) Forced-choice preference: 17/20 preferred dual-mode ($p = .003$).

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."

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 convex hull display helped manage spatial relationships. P11 found that "the toggling system helped" manage stress with many objects, as it provided a way to "handle a large number of objects" 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 expressed frustration in the prescribed-only block because they "couldn't use the collision / support avoider" available in 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 (the minimum distance between any two groups' bounding volumes), 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 in MuJoCo [54] with collision-free motion planning (RRT-Connect, 2 s timeout, 8 grasp orientations).

6.2 Results

Of 100 dual-mode trials, participants delegated at least one group in 69 (69%). The hierarchical workspace optimizer succeeded on all 69 trials (100% feasibility), resolving poses in a mean of 0.02 seconds per trial. To validate end-to-end feasibility, we executed all 200 specifications (100 per condition) on a simulated UR5 robot arm in MuJoCo [54]; the robot successfully completed pick-and-place sequences for both prescribed and delegated groups,

with comparable execution times across conditions (prescribed-only: $M = 125$ s; dual-mode: $M = 132$ s; $p = .19$).

Clearance. The primary H2 signal is inter-group clearance, which determines whether the robot gripper can access each part without collision. 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 ($W = 0$, $p < .001$, 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.

7 Discussion

Our results support the hypothesis that allowing authors to delegate spatial decisions to a downstream executor reduces task completion time, reduces workload, improves robot-executable specification quality, and is preferred by most users.

7.1 Assembly Time Reduction Driven by Fixture-Connection Displacement

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× faster). When two fixtures are assembled together, the connection physically repositions one fixture to align with the other, displacing the sub-assemblies arranged on it. In the prescribed-only condition, confirmation rechecks the displaced arrangement constraints and fails, forcing participants to regrab and readjust every sub-assembly back into tolerance. In dual-mode, those arrangement constraints are delegated and pass regardless of displacement, so confirmation succeeds immediately. P21 captured this experience: dual-mode was "easier in the sense that I do not have to put stuff on the table every time I complete a task." Initial assembly levels (L0) showed a smaller difference (1.4×), consistent with the part-to-part assembly mechanics being identical across conditions. 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%), with the near-total rate for T3 suggesting that for complex assemblies, authors naturally converge toward high delegation. Toggle counts followed the same pattern, confirming that delegation reflects active participant choices rather than default-to-delegated bias.

7.3 Dual-Mode Workflow Improves End-to-End Feasibility

H2 compares the end-to-end workflows, not human versus optimizer in isolation: prescribed-only authoring produces fixed specifications with no downstream optimization, while dual-mode authoring enables an optimizer to resolve delegated poses. The practical finding is that prescribed-only specifications lack gripper clearance: median inter-group clearance was 0 mm (groups placed flush). Human authors, without access to robot kinematic constraints, routinely produce specifications that leave insufficient space for downstream execution. The dual-mode workflow, by enabling delegation, allows the optimizer to achieve 50 mm median clearance. The optimizer improvement over human-placed poses is expected by construction; the contribution is a specification paradigm that makes optimization possible by separating intent (which parts connect) from spatial detail (where they sit).

7.4 Workload and Preference Favor Dual-Mode

Unweighted NASA-TLX total scores (7-point adapted scales, summed across six subscales) were significantly lower for dual-mode ($M = 17.8$ vs $M = 23.5$, $p = .001$), with five of six subscales reaching significance and all six favoring dual-mode. This is consistent with prior work on the relationship between decision authority and operator workload [26]. 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 Constraint Authoring Is Needed

Our work targets the middle of the automation spectrum [36, 13], between fully automated systems that derive constraints from CAD models [8, 59, 12, 4, 43] and manual waypoint programming. Following the function allocation principle [16], the author provides relational intent while the executor optimizes spatial details. This division of decision-making authority aligns with findings that worker satisfaction is driven by team efficiency rather than retained control [17], supporting delegation when it improves outcomes. This division is 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. 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 three design guidelines:

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 observed strategy.

Separate hierarchy encoding from delegation encoding. Our pilot revealed that a single visual channel (nested hull enclosures) could not simultaneously communicate both parent–child relationships and prescribed/delegated state (Sec. 4.1). We resolved this by splitting the encodings: persistent animated lines for hierarchy and on-demand volumes 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 robot (UR5); diverse parts and robot morphologies could test generalizability. Both conditions share the same hierarchy visualization, isolating the IV to delegation choice, though dual-mode necessarily includes

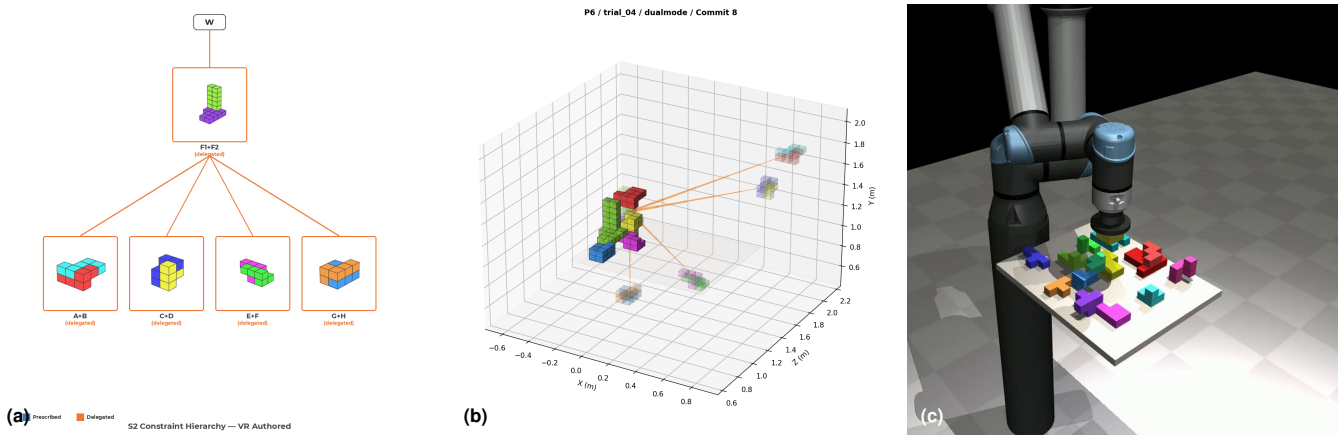


Figure 7: End-to-end pipeline for a T2 trial. (a) Authored constraint hierarchy: the fixture group (F1+F2) and four subassemblies are all delegated (orange). (b) The workspace optimizer resolves collisions among delegated groups (left) to produce a collision-free arrangement (right) while preserving prescribed poses. (c) Simulated UR5 in MuJoCo executing the planned pick-and-place sequence.

delegated volumes as part of the delegation mechanism. This additional visual feedback is a potential confound: dual-mode participants receive richer spatial information (on-demand enclosing volumes) that could independently aid spatial reasoning. However, the primary H1 effect is concentrated at fixture-connection assembly levels where assembling fixtures together displaces sub-assemblies (delegated arrangement constraints absorb the displacement, while prescribed constraints require manual repair), not spatial reasoning aided by volumes. 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. Future work could test whether visual feedback alone (volumes without functional 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 but not grasp synthesis or post-assembly manipulation. Extending to tight-tolerance domains and multi-robot work allocation (assigning delegated groups across manipulators based on reachability) are the next steps.

8.3 Future Directions

Our hierarchical constraint model couples semantic grouping with kinematic parenting: nesting a part under a group both declares a spatial relationship and makes the child move when the parent is manipulated. This coupling creates friction when the author wants to position two related parts without either serving as a fixed reference. Another solution would decouple the constraint graph from the kinematic hierarchy, supporting relational constraints (e.g., "A is adjacent to B") without implying motion propagation. Proxy-mediated interaction [63] offers another avenue.

A related direction is the strictly acyclic hierarchy. Allowing bounded bidirectional propagation (moving a child repositions its immediate parent, but no further) could enable richer constraint semantics for mutually constraining parts. An alternative top-down paradigm in which the author sculpta spatial regions and assigns objects to them could simplify workspace layout tasks whose relevant abstraction is "this zone is delegated" rather than "these parts form a group." A top-down approach would also pair naturally with real-time visual feedback of optimizer solutions during authoring.

9 Conclusions

We presented an XR constraint-authoring interface that separates the author who specifies constraints from the executor who carries them out. Authors build a hierarchical constraint tree through bimanual nesting, with within-group prescribed/delegated modes controlling how much specification authority to retain versus delegate. Animated directional lines and on-demand enclosing volumes communicate hierarchy structure and delegation state. A within-subjects study showed that dual-mode authoring significantly reduced task completion time and workload, and was preferred by most of the participants. Delegation scaled with structure complexity, and the hierarchical optimizer achieved 100% feasibility on all delegated specifications. Because the authored constraints encode spatial relationships rather than executor-specific commands, they can in principle generalize to other downstream assembly workflows.

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. 5

[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 2

[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 1, 2

[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 2, 8

[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 2

[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 3

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

[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 2, 8

[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 5

[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 3

[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 2

[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 2, 8

[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 8

[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 1, 3

[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 2

[16] P. M. Fitts. Human engineering for an effective air-navigation and traffic-control system. Technical report, National Research Council, Washington, DC, 1951. 8

[17] 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 8

[18] 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 3

[19] 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 2

[20] 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 2

[21] 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. 6

[22] 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 2

[23] B. Ikeda and D. Szafir. PRogramAR: Augmented reality end-user robot programming. *ACM Trans. Hum. Robot Interact.*, 2024. doi: 10.1145/3640008 2

[24] 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 2

[25] 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 3

[26] 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 8

[27] 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 2

[28] 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 1, 2

[29] 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 2

[30] 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 2

[31] 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 1, 2

[32] 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 3

[33] 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 2

[34] 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 2

[35] MeasuringU. *10 Things to Know about the NASA TLX*. https://measuringu.com/nasa-tlx/, last accessed on 03/15/2026. 5

[36] 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 8

[37] 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 5

[38] 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 2

[39] 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 2

[40] 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 2

[41] 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 2

[42] I. Quilez. Smooth minimum. https://iquilezles.org/articles/smin/, 2013. Accessed: 2026-03-11. 3

[43] Realtime Robotics. Resolver: Automated motion planning and optimization, 2025. Accessed: 2026-02-12. 8

[44] 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 2

[45] 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 2

[46] 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 2

[47] 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 2

[48] 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 1, 2

[49] 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. 5

[50] 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 2, 4

[51] 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 2

[52] 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 2

[53] 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 2

[54] 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 4, 7

[55] 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 2

[56] 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 4

[57] 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 1

[58] 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 1, 2

[59] 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 2, 8

[60] S.-j. Xu, S. Moore, and A. Cosgun. Shared-control robotic manipulation in virtual reality. In *Proc. HORA*, 2022. doi: 10.1109/HORA55278.2022.9800046 2

[61] 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 2

[62] 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. *arXiv preprint arXiv:2603.01495*, 2026. doi: 10.48550/arXiv.2603.01495 1, 2

[63] 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 9

[64] 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 2

[65] 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 2