DuoZone: A User-Centric, LLM-Guided Mixed-Initiative XR Window Management System

Jing Qian*, George Xi Wang*, Xiangyu Li, Yunge Wen, Guande Wu, Sonia Castelo, Fumeng Yang, and Claudio Silva

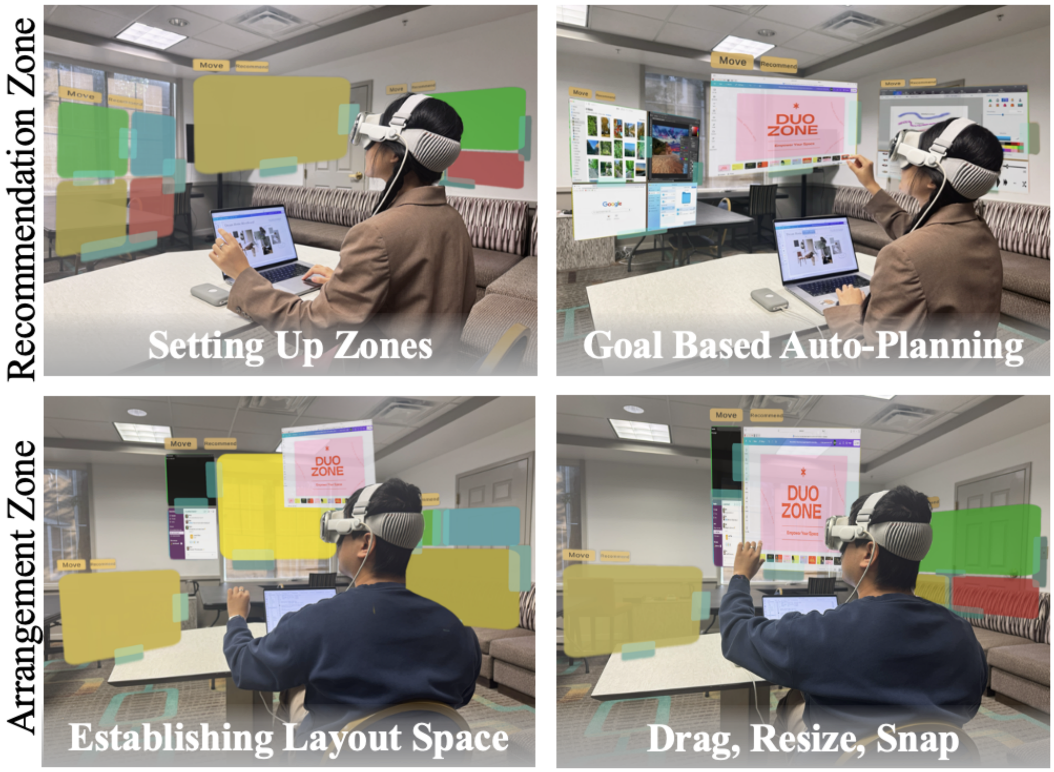

Figure 1: DuoZone uses two spatial configurations (zones) to achieve efficient and low-load XR window management via human-AI collaboration. A Recommendation Zone (top row) uses a cost model with an LLM to automatically recommend relevant applications and adjust layouts. The Arrangement Zone (bottom row) allows users to group, drag, resize, and snap XR applications into user-authored zones.

Abstract— Extended Reality (XR) offers portable, private, and extensive workspaces by extending the virtual workspace beyond desktops. However, current XR window setup and controls remain manual, increasing interaction costs during workspace setup and refinement. We propose *DuoZone*, an AI-assisted scaffolding system that helps users quickly set up an XR workspace and improves speed and efficiency in XR window controls. In DuoZone, users author spatial zones where the automation operates. Given a high-level goal, a large language model (LLM) recommends relevant applications, while a cost model adjusts window ordering and sizing to fit the workspace. An ablation study showed DuoZone provided a more readable and functional workspace setup than an LLM. A user study (N=16) with Apple's Vision Pro further found DuoZone's zone-mediated layout reduced time and workload for window adjustments, while AI-assisted zones reduced initial workspace setup time and workload relative to zones-only setup. These findings provide initial empirical evidence for an "AI scaffolds, user perfects" workflow in short-term XR workspace interaction. More broadly, this work illustrates how user-authored spatial structures can serve as an effective boundary between human intent and AI automation in future collaborative XR interfaces.

Index Terms—XR, mixed-initiative interaction, human-AI collaboration

1 Introduction

The emergence of high-compute-power, lightweight, and high-quality extended reality (XR) systems has opened a new paradigm of portable work, freeing users from the constraints of physical screens [47, 48]. These XR devices offer a large interaction canvas with privacy, allowing the ultimate portability to have an accustomed workspace adapt to mobile working [47, 55] and letting users handle comprehensive and

* *Jing Qian and George Xi Wang contributed equally to this work.*
- *Jing Qian is with Tongji University. E-mail: jqian1590@tongji.edu.cn*
- *George Xi Wang, Yunge Wen, Sonia Castelo, and Claudio Silva are with New York University. E-mail: {xw3617, yw3776, s.castelo, csilva}@nyu.edu.*
- *Xiangyu Li is with miHoYo. E-mail: xiangyu_li@alumni.brown.edu*
- *Guande Wu is with Amazon. E-mail: guandewu@nyu.edu*
- *Fumeng Yang is with the University of Maryland. E-mail: fy@umd.edu*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

complex work outside the conventional office [10, 46]. One crucial benefit of XR working is that it provides an ad-hoc multi-window experience, allowing users to multitask and work on a desktop without carrying multiple physical screens.

However, existing research has identified three main interaction challenges remain for XR working. First, even simple actions such as moving or resizing a virtual window can be challenging in XR [62], causing more effort and physical load on users over time. This becomes more challenging when users attempt to set up their workspace or create a layout in XR. Second, there is a lack of automated ways that focus on facilitating window layout adjustments; existing research, such as SituationAdapt [35], focused on adapting virtual windows to the environment [9, 10] for occlusion avoidance rather than on adjusting sizing or positions. Third, view occlusion caused by a virtual window undermines the effectiveness of XR applications [35], raising safety concerns and limiting users' perception of the physical environment. As a result, automating window layout management is pivotal to reducing users' effort [9], swiftly setting up XR workspaces [45], and potentially improving their focus on the main work [58]. This raises an important question: How can we design an XR window layout management system that improves user performance and reduces user effort?

We propose *DuoZone*, a human–AI collaborative XR window management system that aims to improve the efficiency of layout adjustments, reduce user effort, and simplify initial XR workspace setup through an LLM-plus-cost-model pipeline (Figure 1), enabling a *"user authors, AI scaffolds, user refines"* workflow. DuoZone supports this workflow through two complementary spatial-zone abstractions. **Arrangement Zones** provide user-authored layout structures for snapping, grouping, resizing, and moving multiple windows as a group. **Recommendation Zones** extend these user-authored spatial structures with automation, where an LLM recommends and assigns applications, and a cost model adjusts their ordering and sizing to generate initial workspace layouts for further refinement. Unlike prior optimization-based systems such as InteractionAdapt [10], SemanticAdapt [9], and AUIT [19], which adapt XR layouts through explicitly specified objectives, constraints, or semantic associations, DuoZone uses an LLM to interpret high-level goals and application functionality, together with a cost model for geometric and ergonomic layout adjustment.

To understand and explore the advantages, experience, and drawbacks of the system, we conducted a user study (N=16) with expert designers and engineers (at least 3 years of professional experience). To separate the component contributions, Task 1 compares on-device, native manipulation with Arrangement Zones without AI; Task 2 compares Zones-only setup with the full AI-assisted pipeline (i.e., Recommendation Zones); and a separate ablation isolates the cost model from pure LLM generation. The results indicated that Arrangement Zones significantly improved speed and lowered the cognitive load compared to the condition without using zones. Recommendation Zones also significantly improved the setup speed of XR workspaces while offering a high acceptance rate in automatic app assignment ($> 90\%$), ordering (76.5%), and layout adjustments (82.8%) across two common XR working scenarios (design and programming). Interviews further showed that participants adopted DuoZone's intended workflow in practice, treating AI-generated layouts as editable starting points and selectively refining application choice, ordering, and sizing. Participants generally valued this division of labor and reported feeling in control of the final workspace layout. Finally, the ablation study showed that the cost model produced more readable and functionally appropriate workspace layouts than pure LLM generation.

Our main contribution *is a human–AI collaborative workflow for XR window management that integrates an LLM with a cost model, using user-authored zones as the explicit boundary between human intent and AI automation.* The system is open-sourced on GitHub [1]. Specifically, we contribute:

1. A user-authored zone abstraction for XR window management and workspace planning.
2. The DuoZone system that integrates LLMs' semantic reasoning with a geometric and ergonomic cost model to recommend, assign, order, and size applications within user-authored zones.
3. Empirical evidence from three complementary comparisons: (1) Arrangement Zones reduced layout-matching time and workload relative to on-device window manipulation; (2) Recommendation Zones reduced initial workspace setup time and workload relative to Zones-only setup; and (3) adding the cost model produced more readable and functionally appropriate layouts than pure LLM.

[1] https://github.com/iamgeorge/Duozone-OSS

2 Related Work

2.1 Multi-window management in XR

Window management is a system interface that controls how users view, organize, and interact with on-screen applications by managing display resources [3, 53]. Effective window management supports task management [53, 62]. Virtual window management in XR is more complex than its traditional 2D counterpart because these windows attach to various spatial reference frames, such as the world, body, or user's view [17, 49, 62]. While extra freedom enables expanded workspace, multitasking, and spatial cognition [46–48], it introduces ergonomic issues, limited input precision, and spatial mental overload [15, 17, 46, 62].

Existing methods explored world-fixed [20, 27, 33], head-fixed [15, 51], and body-anchored layouts [34, 56, 62, 64]. Many have adopted spatial layouts as a common approach to window management [16]. However, spatial layouts can overload users when too many visual windows exist [17, 33]. SpaceTop [33] tackles occlusion and z-order confusion by introducing a see-through desktop that distributes windows in depth above the keyboard, blending 2D and 3D interactions. Personal Cockpit [17] reduces visual clutter by organizing windows on a body-centric "cockpit" scaffold around the user. These systems show how immersive, spatially organized workspaces can balance flexibility, structure, and efficiency [47].

Input precision is also challenging [28, 63]. Various systems use different input configurations to achieve better accuracy and speed, such as gaze-assisted systems [16, 46] and embodied interaction [62], and spatial grouping metaphors [52]. Despite the effort, research found that personalization might be needed to support natural interaction in XR [10].

Finally, XR window management can introduce strain due to frequent head movements during multitasking [1]. Inconsistent input mappings and limited contextual control further reduce efficiency [68]. Further, virtual layouts can increase efficiency by reducing application switching time and allowing multiple tasks to remain visible at once [4, 6]. Addressing these gaps requires innovations in multimodal input design, ergonomic adaptation, and intelligent, context-aware spatial window management to realize XR's full productivity potential [48].

Our work builds upon the idea of Personal Cockpit [17] to use spatial anchors for virtual window placing, resizing, and grouping, with the goal of reducing the interaction effort and improving efficacy.

2.2 Workspace productivity with XR

XR productivity refers to how immersive technologies enhance knowledge work by expanding the boundaries of traditional computing [6, 42]. Prior research has explored this in four domains. First, remote collaboration studies examined how immersive platforms like Horizon Workrooms support team communication, highlighting deeper engagement but with issues of discomfort and sub-optimal usability [1, 60]. Second, long-term studies assessed sustained VR work over full weeks, showing potential for deep focus and distraction mitigation. Third, hybrid input methods, such as adapting 2D mice for 3D tasks, combine gaze with touch on tablets to explore efficient arrangements in depth [5, 42, 68]. Finally, work on XR visual guidance on how visual overlays influence task accuracy and cognitive load [50].

There is a trade-off between immersiveness and current technological limitations for XR productivity. Key benefits include expanded workspace and visualization [2, 4, 26, 48], improved focus and engagement [6, 54]. XR also fosters a strong sense of social presence and immersion during meetings over traditional video calls [1], entails portability and privacy [5, 48, 61, 68]. However, several challenges persist, for instance, discomfort, eye strain, and simulator sickness [4, 6, 38, 48]. Technical limitations such as low resolution, narrow field of view, latency, and tracking errors undermine usability [1, 38, 48] and interaction efficacy [4, 54, 68].

2.3 Rule-based constraints for UI adaptation

A substantial body of work formulates MR/AR interface adaptation as decision-making under constraints to reduce cognitive load and improve XR work efficiency [10, 25, 27, 37, 39]. These systems employ rule-based reasoning to modulate visibility [30, 37, 58]. Toolkits generalize these ideas: AUIT lets experts encode objectives and resolve conflicts qualitatively during adaptation design [19], XRgonomics optimizes placements for comfort in 3D UIs [18], and RealityCheck blends virtual and real contexts by compositing real-time 3D reconstructions into VR [27]. Common across this literature is a formalization of UI adaptation as an optimization problem with explicit objectives and constraints (e.g., visibility vs. occlusion, salience vs. distraction, comfort vs. reach), enabling principled trade-offs [9, 10]. Collectively, these approaches advance the field by reducing attentional fragmentation and interaction cost, improving task efficiency, and providing designers with scalable methods that move beyond ad-hoc heuristics [21, 22, 24].

We optimize window management by taking users' physical orientation and interaction history to auto-adjust the layout of virtual windows, reducing the need for participants to perform the tedious work of window resizing.

2.4 LLMs for interface planning, recommendation, and explainability

LLMs can offer interface recommendation that translates users' high-level goals into semantically appropriate interface content, including which applications or UI elements to surface, how they should be grouped, and where they should be placed [12, 36, 40]. Recent systems leverage LLMs to move from descriptive prompts to prescriptive layouts and executable code, such as LayoutGPT [23], Layout Prompter [36], and UI Grammar [41]. This allows users to be more efficient without the need to describe everything in text.

In XR spaces, LLMs can also reason about situated context to provide adaptive and contextually appropriate UI behaviors. They dramatically lower cognitive load, cost for semantic understanding, and enable end-to-end pipelines. SituationAdapt [35] introduces a VLM-based reasoning module that judges occlusion, social appropriateness, and safety while AgentAR [69] uses LLM-based autonomous agents that incorporate users' high-level goals, context, and interaction history to dynamically generate and adapt AR action step-guidance. Our work also uses LLMs to process contextual information to generate adaptive workspaces.

2.5 User agency in human-AI collaboration

In human-AI collaboration, user agency refers to a person's perceived ability to influence and control an interactive system [65–67]. Research shows that direct, manual control fosters stronger agency and embodiment, while high automation can reduce ownership and engagement [67]. However, scaffolded approaches improve perceived agency by keeping users in the creative loop [65].

User agency is linked to how virtual windows should be managed [15,69]. Effective window management requires dynamic, context-aware adaptation that minimizes friction between manual and automated actions [10, 43, 49]. Systems must continuously adjust interface elements to suit the user's context, task demands, and engagement levels. For example, proactive AR agents need to shift their communication mode based on urgency or attention [32, 43, 62]. Poorly aligned UIs can disrupt ergonomics and weaken agency and attention shifts [17, 48].

Several existing works have explored ways to retain user agency in human-AI collaborations. World in Miniature provides an overview that enhances spatial awareness and control [59, 65]. Similarly, embodied authoring enables users to shape and adjust elements in real time, thereby sustaining immediacy and creative ownership [15]. Our implementation is designed to preserve user control by allowing users to author zone locations and refine all AI-generated layouts via natural interactions.

3 Design Goals

3.1 Enabling speed, anchoring, layout, and reducing effort

Using spatial anchors to help improve speed. Arranging XR workspace is crucial for productivity, especially since manipulating virtual windows in 3D can be slower than classic 2D interactions [46]. Rapid layout authoring in XR is difficult due to spatial complexity, depth perception, and limited input precision [56].

Enabling diverse layouts for spatial organizations. The purpose of having layouts is to facilitate complex tasks and multitasking. The layout should be easy to create, adjust, and position in 3D space. It should offer functions to help users group windows to reduce disorientation [15, 53], supporting various application types to customize their sizes within.

Reducing effort improves efficiency. Users often need to adjust the virtual window's size to fit the work with mid-air pointing (i.e., ray casting [28]). However, the large canvas in XR poses more physical demand on all types of inputs. Reducing the effort is key to bringing the 3D experience closer to its 2D counterpart. Further, multitasking within virtual environments escalates the user's mental workload [57] and requires additional physical capacity [31]. One way to minimize the physical effort is by avoiding large arm movements [62]. Meanwhile, simplifying interactions and reducing the number of repeated interactions helps reduce cognitive load.

3.2 User-centered human-AI collaboration

Improve efficiency while keeping users in control. The goal is to use AI for user productivity and experience while avoiding overwhelming them with digital clutter or unpredictable automation. As such, we adopt a mixed-initiative strategy [29]. It is important to let users be in control of the overall layout because window movement without the user's intention causes confusion and requires extra work for the user to update their mental states. Meanwhile, users should easily command AI for tasks and also easily override its suggestions to retain agency [8].

Context is important. XR environments are dynamic; AI adapting to users' context, such as users' physical location, orientation, and interaction state, helps its decision to provide the right information [13]. In our case, context is key for the system to automatically adjust the layout, size, and content of the window.

User's intent can be useful. It is important to obtain detailed intent from the user for the system to automatically recommend useful applications [45]. Using LLMs helps to understand the semantic meaning from the users' intent and reason among different applications to fit the intent [35].

3.3 Dynamic readability suggestions.

System must preserve readability of virtual content for users to digest information. Although larger windows and content improve readability, they also increase navigation effort to navigate in XR, as larger degrees of head and hand movement lead to higher interaction cost. Font sizes should be considered [1] to ensure minimum readability as well as the overall size of the virtual windows.

There is a trade-off between interaction cost (e.g., head and hand movement) and display size. In an XR environment, virtual content closer to the user becomes larger and more readable, but costs more to navigate. As a result, semantically relevant windows (e.g., browser to notes, IDE to terminal) should be placed close to each other for readability.

3.4 Occlusion-free area.

Finally, this design is reserved for direct see-through, allowing users to perceive the physical environment in XR [14, 35]. This area should be easily configurable in sizing and provide a proper warning when interfaces "intrude".

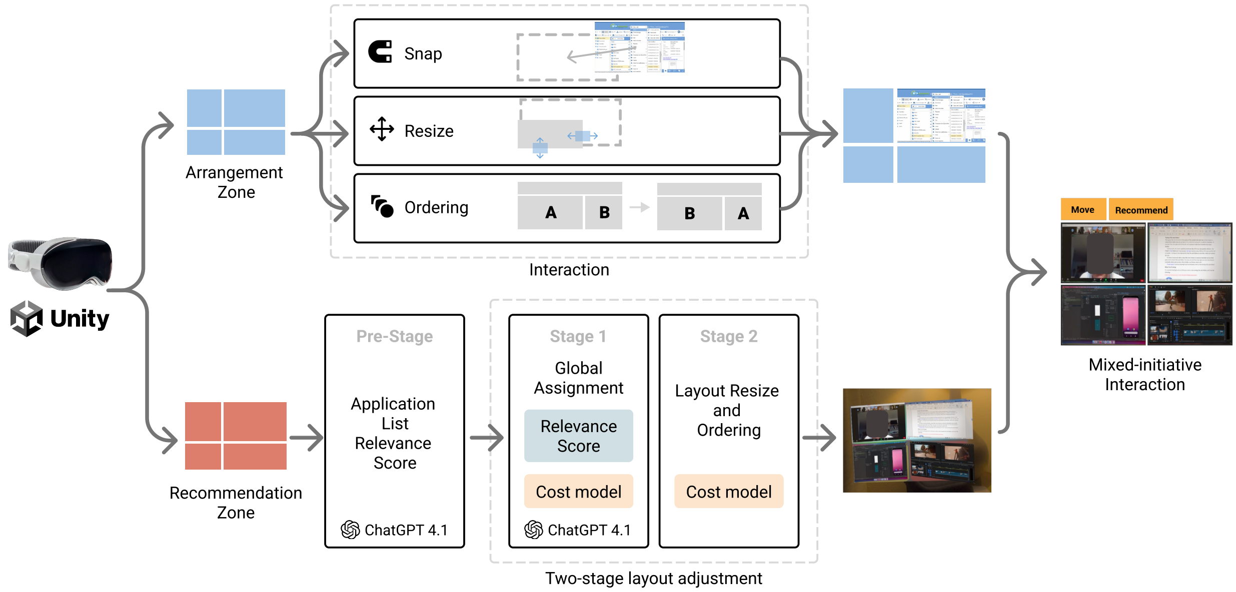

Figure 2: System flow chart to indicate how different spatial configurations contribute to a mixed-initiative experience.

4 DuoZone System

DuoZone aims to achieve efficient window management through customized and human-AI collaborative spatial interactions (See Figure 2). This system is built upon spatial configurations [17] to allow users to customize for different layouts, accommodating types of work (e.g., design, engineering, document organization).

We use **zones** to reference spatial configurations; and zones are interactive layout templates enabling resizing and scaling. Users can fill zones with XR applications (see Figure 3) and move them as a group. These zones make users creators rather than operators of the XR workspace.

We will first present the design and implementation of two zone types: *Arrangement Zone* and *Recommendation Zone*. In recommendation Zones, we describe a multi-stage LLM-based cost model for application recommendation, automatic sizing, and layout adjustment. Afterward, subsections will describe the system setup and data flow.

4.1 Arrangement Zone

As a foundation to support a configurable workspace, Arrangement Zones are translucent, window-like spatial layouts in XR. These layouts function as adjustable containers allowing users to manually adjust their inner cells, overall size, and positions. Inspired by the design of tiled windows [7], we designed six layout templates that are commonly used in modern operating systems to distribute space within a zone. These six templates cover basic horizontal and vertical tiling windows and are not inclusive. In practice, users can construct more complex layouts from these templates and assemble them into more templates based on the needs. Spatial areas covered by Arrangement Zones will provide semi-automatic organization; other spaces in XR will remain fully manual user control. This setup allows users to actively plan for their workspace layout with their intended needs and functions.

4.1.1 Interaction.

Interacting with the Arrangement Zone follows the traditional drag-and-drop metaphor, supporting a set of interactions commonly used (e.g., snapping, alignment, grouping). The goal of adopting such a design is to improve the efficacy of common micro window operations (DG1), see Figure 3 for how each interaction works. These interactions are implemented with Apple's PolySpatial library in Unity. The library does not access raw continuous gaze, as visionOS exposes only a system-mediated interaction pose at the pinch event (SpatialPointer) such as gaze-targeted selection plus hand pinch. The event are used for interface state changes (e.g., drag start, drag end, taps/clicks, hover). For example, once the user drags a window while hovering on a cell, it highlights to indicate a change of state, indicating a snapping follows.

4.1.2 Occlusion-free template

We designed a type of Arrangement Zone that allows users to label a "void" area in XR. This area is directly see-through for physical environment visibility. It is designed as a 1x1 layout, and the user can move and adjust the size upon placement. When other zones intersect, a red contour encompassing the zone will be visible, alerting the user of the restricted placement area.

4.2 Recommendation Zone

We designed another type of zone focused on providing contextual automation. This gives the user an alternative mentality to "outsource" some of the management burden. These zones allow users to keep autonomy and decide what not to be automated, reducing the risk of loss of agency during a human-AI collaboration.

The core challenge of the AI recommendation has two parts: 1) to decide what virtual window app to put in which cell of multiple zones, and 2) how to determine the size, order, and layout of zones. We do not want to change the location of user-authored zones for retaining their spatial mental model [11], but allowing for size changes to optimize interaction cost in XR.

We begin with users who created K Recommendation Zones in XR; each zone Z_k $(k = 1, \dots, K)$ is characterized by:

- **Layout** $\tau_k \in \mathscr{T}$, where $\mathscr{T} = \{1{\times}1, 2{\times}2, 1{\times}2\text{v}, 1{\times}2\text{h}, 2{\times}1\text{v}, 2{\times}1\text{h}\}$, see Figure 4 for details.
- **Dimensions** (W_k, H_k) as width and height in meters
- **3D transform** (p_k, o_k) where $p_k \in \mathbb{R}^3$ is position and $o_k \in SO(3)$ is orientation relative to the user
- **Cell set** $\mathscr{C}_k = \{c_{k,1}, c_{k,2}, \dots, c_{k,n_k}\}$ where n_k is the number of cells determined by τ_k

4.2.1 Pre-Stage: LLM-based Application Relevance Prediction

Our system uses ChatGPT 4.1 for semantic and goal-aware reasoning from user's high-level goal. When the user triggers an AI recommendation, we use Meta's Wit for voice-to-text and provides LLM with the transcribed text. Users can edit the text via our interface if needed. Based on this goal (G), LLM will output a set of recommended virtual window applications $\mathscr{A} = \{a_1, a_2, \dots, a_N\}$. Meanwhile, an accompanying set of relevance score $\mathbf{r} = \{r_1, r_2, \dots, r_N\}$ where $r_i \in [0, 1]$ represents the predicted likelihood that application a_i will be used.

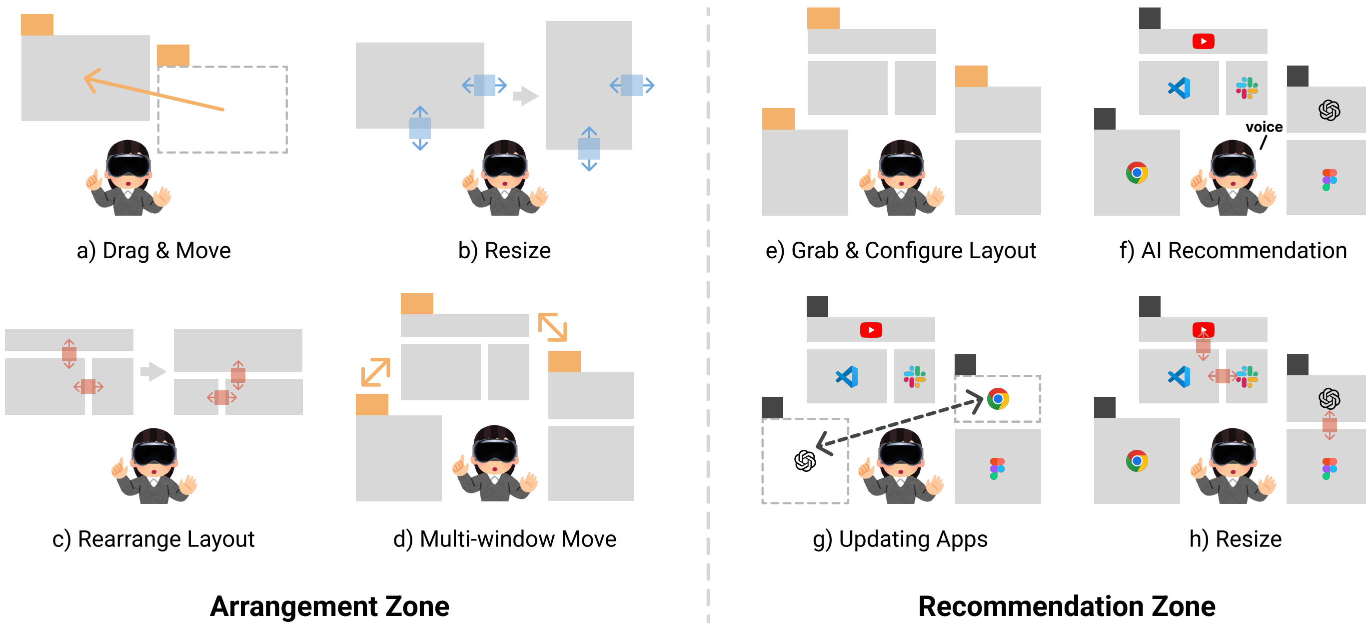

Figure 3: Interactions in DuoZone's system. In the Arrangement Zone (a–d), users directly drag, resize, rearrange, and move multiple windows to create personalized layouts. In the Recommendation Zone (e–h), users can voice requests for AI-generated layouts. The AI will automatically recommend apps and fine-tune window sizes. Together, these modes balance user control with AI assistance for efficient workspace organization.

4.2.2 Constraints

We define the following constraints for the further two-stage interaction models. For sizing constraints, it is important to note that: 1) Each cell contains at most one application, but could be empty; 2) The number of applications assigned to zone k cannot exceed its cell count; and 3) the layout's dimension must be within the zone dimensions. For readability, each cell must maintain a minimum angular size to ensure text readability. Following Microsoft's MR Typography guidance for readability [44], we use 0.5 *degrees* of visual angle as a minimum legibility threshold for near interaction.

4.2.3 Interaction Cost Model

To reduce XR setup effort and minimize manual adjustments (Section 3.2), we implemented a two-stage cost model to support efficient workspace layouts by optimizing application adjacency, placement, and functional sizing (e.g., larger landscape-oriented windows for IDEs). We define the following cost signals.

F_{ij}: **Pointing Distance** measures the Euclidean distance between applications a_i and a_j from cell centers. If both are from the same cell we use, $F^k_{ij} = \|\mathbf{c}^k_i - \mathbf{c}^k_j\|$, and otherwise, $F^{\text{Zone}}_{ij} = \|\mathbf{p}_k - \mathbf{p}_\ell\|$.

M_j: **Hand Movement Distance** measures the Euclidean displacement of the hand during direct manipulation, from $\mathbf{c}_{\text{prev}}$ to $\mathbf{c}_{\text{current}}$ between a *pointerdown* and *pointerup* event; because Vision Pro supports continuous direct manipulation without requiring users to lift their hand, this movement may be nonlinear relative to zone-to-zone distance.

H_j: **Head-Turn Angle** measures the angular displacement required to bring application a_j into the user's forward view, defined as the angle between the forward viewing vector $\mathbf{v}_{\text{forward}}$ and the direction vector $\mathbf{v}_j$ from the user to the application center: $H_j = \arccos\left(\frac{\mathbf{v}_{\text{forward}} \cdot \mathbf{v}_j}{\|\mathbf{v}_{\text{forward}}\| \|\mathbf{v}_j\|}\right)$.

In practice, we use a list of dictionaries to store these cost signals in a time sequence. The list also contains a normalized $\widetilde{F}_{ij}, \widetilde{H}_j$, and $\widetilde{M}_j$ that are determined from the zone's layout and configuration.

Stage 1. We use the reasoning capabilities of ChatGPT-4.1 to recommend a global application assignment across all cells and zones. This approach considers nuanced trade-offs, contextual relationships between applications, and users' potential costs that could be difficult to encode in optimizing objectives. We first compute a cost matrix using initial layout parameters Θ^{init} (layouts with default splits or sizes given by the user):

$$C_{i,k,j} = \sum_{\ell \in \mathscr{A}_{\text{prev}}} \left[r_i r_\ell P_{i\ell} \cdot c^{k,j}_{i\to\ell} + r_\ell r_i P_{\ell i} \cdot c^{k,j}_{\ell\to i} \right] \tag{1}$$

Where $r_i r_\ell$ denotes the joint relevance of applications (obtained from previous LLM call) i and ℓ, representing the likelihood that both applications will be used together. Furthermore, $P_{i\ell}$ is the transition frequency from application i to ℓ and $c^{k,j}_{i\to\ell}$ is the realtime transition cost from application i (hypothetically placed in cell (k, j)) to application ℓ. The transition-frequency matrix $P_{i\ell}$ is maintained separately for each user and estimated from that user's application-transition history. When no prior interaction data are available, $P_{i\ell}$ is initialized uniformly as a neutral cold-start prior. Finally, $\mathscr{A}_{\text{prev}}$ is a set of previously assigned applications.

The instantaneous cost is computed as following. Because F, H, and M use different physical units (e.g., meters and degrees), we normalize each term to a $[0, 1]$ scale using ergonomic min–max ranges. A heuristic piloting by two authors suggested that calibrating different weights to individual head, hand, and pointing with limited interaction data could lead to overfitting; therefore we opt for equal weights for the three normalized terms:

$$c^{k,\ell}_{i\to j} = \begin{cases} \lambda_f \widetilde{F}^k_{ij} + \lambda_h \widetilde{H}^k_j + \lambda_m \widetilde{M}^k_j & \text{same zone } k \\ \lambda_f \widetilde{F}^{\text{zone}}_{ij} + \lambda_h \widetilde{H}^{\text{zone}}_j + \lambda_m \widetilde{M}^{\text{zone}}_j & \text{different zones} \end{cases} \tag{2}$$

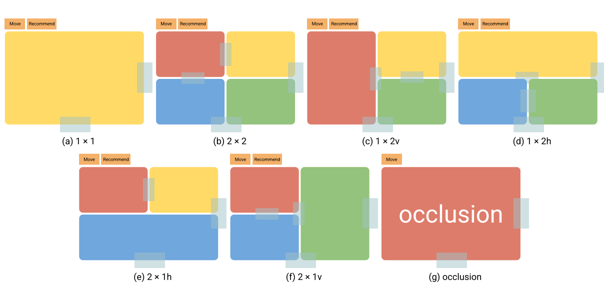

Figure 4: Extending the tiled window convention, we use six different layout templates as 3D spatial anchors to support window resize, ordering, snapping, and grouping. An occlusion template placed in XR space allows users to see-through without virtual information blocking the view.

LLM Prompting Decisions. Along with the cost matrix described above, we construct the prompt with the following data: 1) recommended application list with relevance scores from the first LLM call; 2) user created zones with their layout, sizes, and all occupied cells; 3) the cost matrix $C_{i,k,j}$; 4) readability constraints; 5) high-level goal G for context. The LLM's output considers multiple factors simultaneously, such as relationship and appropriateness along with numerical data. For example, LLM might agree that "emails and calendars should be placed adjacent" even if the pure cost matrix suggests otherwise.

LLM Output Parsing. We use fixed prompts that require structured JSON outputs for scenario inference and application recommendation. The live prototype parses responses directly; malformed outputs or request failures return an error without automatic retry. Full prompts and implementation details are provided in the open-source repository [2].

Stage 2. Our preliminary testing found that using prompts alone yielded inconsistent results in resizing the zone's layout or size, even if we provide information for users' distances or angular distances. As a result, after we obtain the assignment results from stage 1, each zone k independently optimizes its internal layout parameters. Let $\mathscr{A}_k = \{a_i : \sigma^*(a_i) \in \mathscr{C}_k\}$ denote applications assigned to zone k. The goal here is to find a layout configuration θ where:

$$\theta_k^* = \arg\min_{\theta_k \in \Omega_{\tau_k}} \left[\mathscr{C}_k^{\text{local}}(\theta_k \mid \sigma^*) + \lambda_s \mathscr{S}_k^{\text{local}}(\theta_k \mid \sigma^*) \right] \tag{3}$$

Here θ represents the point (w_0, h_0) within a zone where horizontal divider crosses w_0 and vertical divider crosses h_0. Each zone is resized locally based on its configuration and layout. To do that, we follow:

$$\mathscr{C}_k^{\text{local}}(\theta_k \mid \sigma^*) = \sum_{i,j \in \mathscr{A}_k} r_i r_j P_{ij} \cdot c_{i \to j}^{\text{intra}}(\theta_k) \tag{4}$$

where:

$$c_{i \to j}^{\text{intra}}(\theta_k) = \lambda_f \widetilde{F}_{ij}^k(\theta_k) + \lambda_h \widetilde{H}_j^k + \lambda_m \widetilde{M}_j^k \tag{5}$$

We then calculate the local size-relevant term, which encourages allocating more space to higher-relevance applications within zone k.

$$\mathscr{S}_k^{\text{local}}(\theta_k \mid \sigma^*) = -\sum_{i \in \mathscr{A}_k} r_i \cdot A_i^k(\theta_k) \tag{6}$$

A final step after finding the results of the most optimal θ_k is to conditionally scale-up the entire zone if the smallest cell in the zone does not match the readability constraint. The scale-up rule is a straightforward geometric adjustment to enforce the 0.5-degree typography introduced in Sec 3.3. The system calculates the angular size of the smallest application (cell) within a zone and applies a uniform scale-up until the threshold is met. Finally, the layout can be updated using W_0 and h_0 to the following, see Figure 4 for visualization.

4.3 Ablation Study

To better understand the cost model's immediate benefit (*Cost*), we conducted an ablation study comparing to simple LLM recommendations (*LLM*). The *LLM* condition used ChatGPT 4.1 to recommend relevant applications upon a rater's request. The average round-trip response time was 4.13 seconds on a Wi-Fi condition over 160 paired workspaces. The LLM directly placed, ordered, and sized the applications into the rater-defined zones. The *Cost* condition followed our proposed pipeline for recommendation and sizing adjustments.

We evaluated these two conditions across 2 different workspace setups (i.e., design, engineering) and rated the results based on human judgment of relevancy (whether the recommended apps are needed for the current setup), readability (if apps are resized enough to read), and functionality (app sizes were appropriate for their functions) using a five-point Likert scale. Two independent raters with moderate inter-rater agreement (Cohen's kappa = .61) evaluated 160 paired workspace setups (LLM + cost vs pure LLM) using paired Wilcoxon signed-rank tests. The results indicated that the application recommendations were both high and similarly relevant (Cost: $M = 4.18, SD = 0.62$; LLM: $M = 4.11, SD = 0.69$); however, the app sizing and placement were less satisfying ($W = 342.0, p < .001$) in the *LLM* condition ($M = 2.76, SD = 1.49$) compared to the *Cost* condition ($M = 4.06, SD = 0.62$). Readability for the *LLM* condition($M = 2.79, SD = 1.52$) also scores significantly less ($W = 245.0, p < .001$) than the *Cost* condition($M = 3.98, SD = 0.66$), as *LLM* often sized applications too small to read the content in the XR environment without clear reasons.

[2]https://github.com/iamgeorge/Duozone-OSS

5 User Study

To further understand DuoZone's efficacy and its AI-assisted scaffolding for XR window management, we conducted an empirical study with expert users who worked under multi-window situations in their daily lives. We used a within-subject design and devised two primary tasks. These tasks elicited users' performance and the cognitive and physical effort of our system. The study primarily evaluated short-term workspace setup, window management operations rather than long-term downstream productivity.

We asked the following research questions: 1) Whether DuoZone improved the interaction speed and users' cognitive load; 2) Whether DuoZone reduced the number of manual adjustments to virtual windows; 3) In what ways DuoZone's recommendation could be useful? and 4) How did DuoZone's recommendation compare to users' manual setup?

5.1 Apparatus

We used an Apple Vision Pro (1st-generation) as the XR headset that connected to our *DuoZone* client. The client was implemented in Unity using Apple's PolySpatial integration. A dedicated host equipped with an NVIDIA GeForce RTX 4050 GPU handled real-time AI assistance, cross-process communication, and threading. The headset and server communicated over the local network for low-latency data transfer.

5.2 Participants

We recruited 16 participants (8 males and 8 females) between 21 to 39 years old (M=26.7, SD=5.5). They all had at least three years of professional working experience and used multi-window configuration in their daily jobs. We used snowball sampling and sent digital flyers on social media and direct messaging tools. All but one participant had prior experience with XR interaction, and had experienced with virtual windows before this experiment.

5.3 Conditions

Our goal was to isolate the incremental value of the proposed components under a controlled input. However, existing related systems were either not directly comparable or not functionally equivalent in our settings, as end-to-end SOTA systems would introduce confounding variables. As a result we used a strong and common interaction metaphor (Vision Pro's native window interaction) as the starting for internal validity.

Baseline. The baseline differed by task, both without AI engagement. In Task 1, the baseline was Vision Pro's native window manipulation (Without AI). In Task 2, the baseline was user-created Arrangement Zones(Without AI). For **DuoZone** conditions, both tasks involved user-authored zones, the only difference was the AI engagement. Task 1 used Arrangement Zones (without AI) and Task 2 used Recommendation Zones (with AI).

Balancing and Order. To reduce the learning and ordering effects, we counterbalanced the conditions using an alternating design for Task 1. In Task 1, we used a pre-generated table that randomized and balanced both the direction and layout across five directions, ensuring that the total number of layouts and directions tested was equal. In Task 2, we always started with the baseline condition because participants' decisions on app choices and arrangements might be affected by LLM (DuoZone condition), and the reverse did not hold. While this ordering introduced a potential carryover effect, we considered it preferable to the alternative.

5.4 Tasks

5.4.1 Task 1: Layout Matching

Participants performed a precise layout-matching task using the DuoZone and Baseline conditions (Figure 5) without AI involvement. There were 15 trials per condition and 30 total. In each trial, 3–4 application panels (e.g., browser, notes, chat) appeared in front of the user. A holographic image then appeared with a specified workspace configuration, representing the assembly goals. The participants needed to move each application panel to match the holographic image until they were satisfied with the result. The holographic image would appear in any of the five directions (left, top-left, top, top-right, right) where the participant was at the center. The overall appearance frequency across the five directions was balanced throughout the study.

5.4.2 Task 2: Workspace Construction

The goal was to set up a workspace they would normally use to design a graphic poster or to code scripts for an interactive demo (see Figure 5 (h)). In the baseline, participants selected apps from a palette of approximately 20 applications relevant to their goal, and placed, ordered, and sized them into Arrangement Zones, similar to the first task. In DuoZone conditions, participants issued a high-level voice command (e.g., "set up for coding a web game") to initiate LLM after they set up the empty Arrangement Zones. The LLM then placed, ordered, and sized the application based on our proposed method. The participant was then given time to adjust the workspace until satisfactory.

5.5 Procedure

All sessions were conducted in a quiet, well-lit indoor lab space with ample open area to permit natural head and hand movement. Participants performed the study seated to simulate a working environment. After lens adjustments and headset fit, participants wearing Vision Pro would follow an experimenter's instructions for tasks once they sign the consent form. They would perform Task I first followed by Task II. Because the two tasks had different goals and this order minimizes the learning effect. At the end of each task, participants were asked to complete a short survey and a NASA-TLX form to understand their cognitive load. After the study completed, the experimenter conducted a semi-structured interview to understand high-level questions such as agency, efficiency, potentials and drawbacks.

5.6 Data Collection

Timed-performance: We wrote a script capturing the time difference between a pointer-down to pointer-up event for Task 1 to detect interactions. For task 2, the time was calculated in total number of seconds to complete.

Cognitive load: We used a digitized NASA-TLX form that follows the original 0 to 100 TLX score. Each of the six category questions was designed as a 21 tick slider, with the minimum increment of 5.

AI Recommendation Acceptance: We measured AI acceptance from the following dimensions: 1) recommended application acceptance, 2) ordering and adjacency, and 3) sizing. A script was used to record button responses from whether a user accepts or declines application recommendations in each cell of a zone, and recorded the number of re-ordering and sizing adjustments following LLM's attempt. Data were also cross-validated by two separate authors using the Vision Pro's first-person recordings. The final count was combined with any disagreement resolved through discussion.

Semi-structured interview: We collected semi-structured interview questions at the end of each task. We recorded both the audio and video of each experimental session and used YouTube's auto-transcription function to extract transcription from the videos.

In total, we collected 480 trials from 16 participants for Task 1. All participants completed two different conditions in Task 2, constitute 32 workspace setups. We recorded a total of 32 videos (one third-person and one for first-person) with each about 1.5 hours.

Figure 5: (a–f) Participants performing interaction tasks while wearing Vision Pro headset. (g) Task 1 interface, where participants were instructed to drag and place applications into designated target locations. (h) Task 2 interface, where participants configured multiple applications into a workspace with or without the help of LLM.

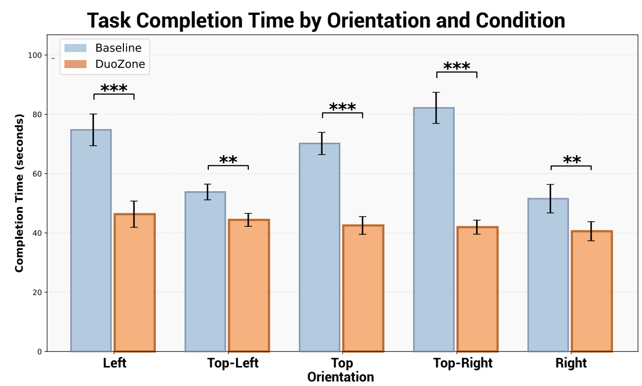

Figure 6: DuoZone significantly reduced completion time across five different orientations.

6 Results

6.1 Timed-performance

We log-transformed Task 1's completion time and a paired t–test revealed a significant main effect of *Condition* ($t(15) = 6.22$, $p < 0.001$). On the original scale, the geometric–mean time ratio of DuoZone over Baseline was 0.655 with a 95% CI of [0.566, 0.757], indicating that *DuoZone* performs 34.5% faster performance on average.

Within each *condition*, we compared all pairs of orientations using two–sided paired t–tests on the log-transformed time (see Figure 6); p–values were Bonferroni–adjusted within condition (10 comparisons per family). In *Baseline*, Orientation Top-Right tended to be slower and Right faster relative to others. In *DuoZone*, no orientation pairs performed significantly different from others.

For Task 2, the normality assumption was not violated for baseline ($p = 0.398$) and DuoZone ($p = 0.939$) using the Shapiro–Wilk test. A two-tailed paired-sample T-test showed that users in DZ-2 condition is significantly faster ($t(15) = 3.23, p = 0.006$) than that of the baseline, with DuoZone scores an average time of 168.3 seconds ($SD = 67.5$) when user finished the setup and the baseline scores an average of 218.1 seconds ($SD = 73.1$), as shown in Figure 7.

6.2 Cognitive Load

The TLX's subscale ratings did not meet the normality assumption using Shapiro–Wilk test ($p < 0.05$); therefore nonparametric tests were used for this analysis (see Figure 8).

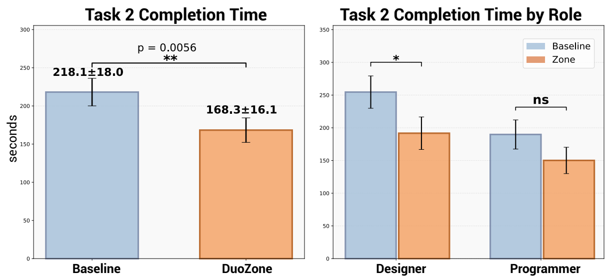

Figure 7: DuoZone shortened workspace construction time. The left chart showed the overall completion time for the workspace construction task.

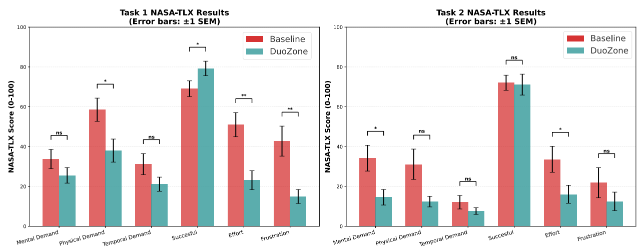

Figure 8: Comparison of NASA-TLX subscale scores between Baseline and DuoZone conditions for both Task 1 and Task 2.

For Task 1, a Wilcoxon signed-rank test revealed a significant difference in overall cognitive load ($p = .006$), with the *DuoZone* condition ($M = 33.30$) reporting lower scores than *Baseline* ($M = 47.71$). Post-hoc two-sided Wilcoxon signed-rank tests showed that *DuoZone* significantly reduced Physical Demand ($W = 22.0, p = .017, r = .59$), Effort ($W = 16.0, p = .007, r = .67$), and Frustration ($W = 9.0, p = .004, r = .72$) relative to *Baseline*; meanwhile, performance is significantly improved for DuoZone ($W = 23.0, p = .035, r = .53$). No other subscales reached significance. See Figure 8 for details.

For Task 2, a Wilcoxon signed-rank test found a significant difference in the overall cognitive load ($p = .004$), with *DuoZone* condition ($M = 22.38$) less than *Baseline* ($M = 34.16$). Post-hoc two-sided Wilcoxon signed-rank tests showed that the AI-assisted DuoZone condition significantly reduced Mental Demand ($W = 24.0, p = .041, r = .51$) and Effort ($W = 27.5, p = .036, r = .52$) relative to Zones-only. No significant differences were identified for the remaining subscales.

6.3 Acceptance of AI recommendations

High-level Goal-related application recommendation: On average, each user created about 4 zones, with AI recommending 8 virtual apps per workspace setup. Within these recommendations, users rejected on average 0.68 apps ($\sigma = 0.87$), resulting in an average of *91.7%* acceptance rate ($\sigma = 13.7\%$). See Figure 9 for details.

Cost-model driven layout sizing and ordering: After the cost-model adjusts the layout of the zones, the average user further adjusted 0.68 out of 4 zones' layout, or accepted 82.8% of AI layout suggestion (see Figure 9). In terms of ordering and adjacency, the average user accepted 76.5% of automatic ordering. Meanwhile, manual layout scaling accounts for 37.5% (62.5% acceptance rate).

6.4 Qualitative Analysis

We coded the survey and interview data using both open and axial coding. Two coders separately coded the data followed by a cross-check to enhance reliability.

6.4.1 Preferences were dependent on users' need for precision or stability.

Baseline was preferred when precision was required or when on-the-fly and quick adjustments are needed (P9, P12). It is also easy to quickly zoom in and out on a single virtual window for design situations (P13). Moreover, participants appreciated independent control over each virtual window, as adjustments in the baseline condition do not affect others (P6, P15, P16), or for highly customized layouts (P8, P16).

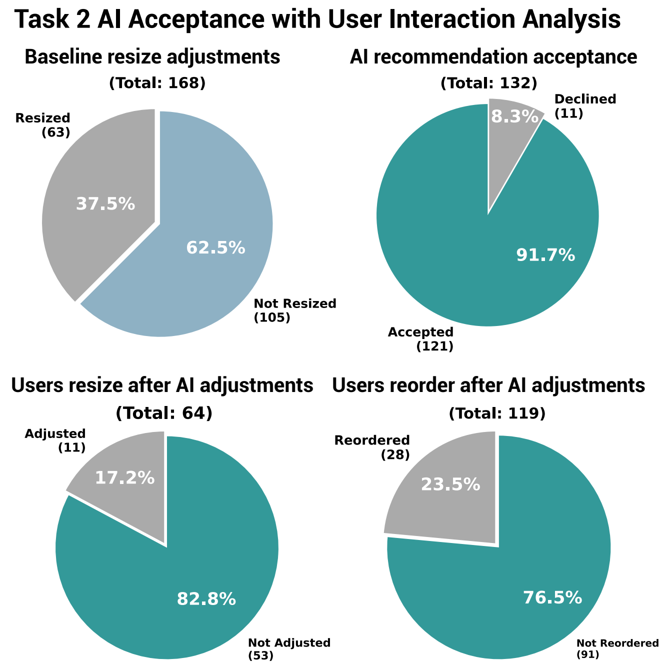

Figure 9: Visualization of user interaction and AI recommendation acceptance in Task 2. The charts show how users accepted or adjusted AI-generated workspace layouts, with high acceptance of recommended applications and layouts and relatively few post-adjustments for resizing or reordering compared to the manual baseline.

Preference for Arrangement Zones was more towards stability, organization, and consistency (P1, P4, P6, P8, P10, P11, P12). Many brought up the potential for daily multitasking, especially tasks with consistent layouts (P3, P5, P9, P14, P15). Many participants opted to place the Arrangement Zones first and then drop in virtual windows and fine-tune the zone's layout (P7, P9, P12, P10, P11), as they believe this would reduce setup costs.

6.4.2 Participants feedback on AI acceptance

Overall, participants reported similar satisfaction ($W = 73.50$, $p = .927$) for Manual ($M = 3.81$, $SD = 0.88$) and DuoZone ($M = 3.69$, $SD = 1.06$) conditions.

Recommendations create usable, intention-aligned layouts. Overall, participants were positive about the virtual applications recommended to them due to both high relevance (P9, P11, P13, P15, P16) and reasonable workspace layout and sizing. P11 found that the recommended apps "perfectly align with my intention...exactly what I want", and P3 and P15 affirmed that AI's coarse layout "matched my thinking" and was great for quickly getting a usable scaffold when they "have no idea". Even when unnecessary apps were included, participants felt the necessary components were present (P13, P15, P16).

Primary apps are well-centered and other clustered apps support streamlining workflow. Users highly valued that AI placed the right app in the center of their vision (P7, P8, P16), for example, an IDE for coding or a design tool. Participants also noted that the AI placed related supporting applications near each other, which facilitates the workflow and reduces searching effort. P7 said that "...placing File and Note together made it easier to access and view"; "Gmail next to a chat was useful because both information sources required for decision-making are visible simultaneously." (P8).

AI layouts were well-received but could benefit from personalized fine-tuning. Participant (P1) also noted that "AI doesn't know my personal work habits...working is very personalized, " and P7 said that "without my prior experience as input, it's hard to infer...". P16 added that "AI gives the most size-relevant, but I needed to adjust the details". These findings suggested that participants accepted the general structure but still needed to fine-tune the app's size to match their habits. While the overall text readability was good (P13, P2), participants tended to adjust zones wider horizontally and flatter vertically. As P10 brought up "Photoshop is too narrow ", P15 brought up "A PDF reader can't be that narrow",and P6 said that "A squashed browser was hard to use...". These indicated that specific sizing requirements may be different for each application.

6.4.3 AI scaffolded, user perfected

Despite imperfect sizing, ordering, and app placement, participants were generally positive towards a collaborative approach. P3 said that "AI's layout matched my thinking, but some sizes weren't perfect; I could fine-tune on that framework", while P12 mentioned that AI can "just give me a framework, then I refined what i truly needed". P15 further brought up that " AI did the layout fast; I'll modify afterward. This was convenient.". Overall, acceptance was high on the framework-plus-selective-adjustments approach to retain control over critical details.

6.4.4 Manual adjustment was precise but time-consuming.

Participants praised the high level of customization and flexibility in manually adjusting the virtual windows, with the ability to adjust individual apps freely (P5, P2, P8, P16). Manual adjustment also allowed them to precisely place supplemental materials such as documents, image assets library, or notes to the side for easy access (P1, P7, P15).

The downside to the manual adjustment is its operational complexity, as participants had to "manually drag and adjust every application" and felt it was a "time-consuming process" (P5). Moreover, as more applications pile into the XR space, several participants reported overlapping issues when trying to form layouts, causing selection and visual occlusion issues (P15).

6.4.5 Occlusion was vital for awareness and multitasking

Participants described occlusion as an essential element of XR interaction (P13, P14, P15, and P16). They emphasized Occlusion Zone's partial transparency sustained peripheral awareness without disrupting focus (P15). For example, they would "place it on the desk view to see my hands and computer". P15 framed occlusion as critical for social reasons: "If my boss walked in while I'm slacking, I needed [to see]it near the door". P16 highlighted the value of environmental connection, "let me know when someone greets me". Collectively, participants articulated that occlusion operated as a core mechanism for maintaining situational awareness, supporting multitasking, and reinforcing the sense of control.

7 Discussion

7.1 DuoZone Improved Interaction Speed and Reduces Cognitive Load

We found that DuoZone significantly improved temporal efficiency and reduced cognitive workload (RQ1). In Task 1, completion time decreased by 34.5% compared with the baseline ($p < .001$), and setup time was shorter in Task 2 ($p = .006$), suggesting that the zone-mediated metaphor and AI-assisted scaffolding benefit layout interaction and setup in XR.

DuoZone reduced manual adjustments (RQ2) with a high acceptance rate of 82.8% for layout structure and 76.5% for ordering. Interviews suggested that users perceived DuoZone's AI output as a "scaffold rather than a final product " and employed a "two-stage mental model" where "AI scaffolds, user perfects". They also admitted that DuoZone provided a great way to "quickly get a usable scaffold" when they "had no idea." (P10,P12,P13), reflecting the benefits of the AI-assisted collaborative workspace setup.

DuoZone's AI recommendations were well-aligned with user intentions (RQ3). The 91.7% acceptance rate and 82.8% layout acceptance rate revealed that the system generally produced usable application selections and coarse layout structures, while the lower 62.5% manual sizing acceptance highlighted the continued need for personalized refinement.

7.2 Comparison Between DuoZone and Manual Setup

DuoZone's results were highly satisfactory to users (RQ4). One reason was that the system cut setup time by 50 seconds and reduced effort. Interviews revealed that users did not perceive automation as compromising quality or expressiveness. When comparing the two, the preference scores were similar across DuoZone and Baseline, reflecting a deeper trade-off between flexibility and structure. Manual setup afforded maximum freedom but imposes high costs, whereas DuoZone imposed light structural constraints but less personalization.

7.3 Potential Use in Task-Specific Workspaces

Recommendation Zones translated semantic intent (e.g., "coding a web game") into actionable layouts that pre-structured the workspace. Participants valued this as a "usable starting point". They (P11,15) mentioned that DuoZone could quickly stage all resources in XR for tasks with low familiarity. Participants (P12, 16) also emphasized that LLMs could internalize local application norms in cross-organization collaborations where tools are unfamiliar across companies (Gmail over Outlook or Slack over Discord). When a stable "home base" is unavailable, AI mediation becomes the most convenient.

7.4 Perceptions of Control and Trust

Participants' reflections suggested that DuoZone's architecture helped them maintain a sense of control in the generated layouts. Participants mentioned trust because of goal alignment and reasonable spatial layouts. The Recommendation Zone was repeatedly described as "perfectly aligned with my intention," with users appreciating how the AI placed primary tools centrally and grouped related applications proximally (e.g., "File and Note together"). These intelligent placements reduced visual search and improved workflow coherence, leading users to describe the AI as "reliable" and "thoughtful."

8 Limitation

The work has several limitations. First, its reliance on LLMs can lead to potential delays. Second, AI recommendations lack a deeper understanding of users' work habits. Third, occlusion-free areas are evaluated qualitatively, as they are not the main focus of this work but are used as extensions of the Arrangement Zone. Fourth, since Task 2 used a fixed order, the design cannot eliminate order and carryover effects. As such, Task 2 supports the usefulness of AI-assisted scaffolding rather than providing definitive causal evidence independent of order effects. While the user study is conducted on a small scale (N=16), the within-subject design yielded 480 Task 1 trials and 32 complete workspace setups from expert users. Yet the effect on long-term, subsequent productivity remains in future work. Finally, our evaluation used native Vision Pro manipulation rather than end-to-end SOTA systems, which improved internal validity but limits broader comparative claims.

9 Conclusion

We present DuoZone, a human-AI collaborative window management system to improve interaction performance and lower workload. The system established a *user-authored, AI-assisted* workflow via two types of zones for efficiency. The **Arrangement Zone** enables spatial anchors for snapping, resizing, and group positioning; and the **Recommendation Zone** uses users' high-level goals and a cost model to automatically recommend apps, adjusting layout sizing and ordering. A 16-user empirical study found that DuoZone significantly reduces cognitive load and time, while automatically recommended apps have an over 90% acceptance rate. The work demonstrated an automated approach to XR window management and explored an acceptable starting point for collaborative XR workspace interactions.

Acknowledgments

This work was supported in part by the DARPA PTG program and NSF grant OAC-2411221. The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the views or policies of these agencies. Generative AI was used solely to improve the writing and grammar of the manuscript.

REFERENCES

[1] K. Abramczuk, Z. Bohdanowicz, B. Muczyński, K. H. Skorupska, and D. Cnotkowski. Meet me in vr! can vr space help remote teams connect: A seven-week study with horizon workrooms. *International Journal of Human-Computer Studies*, 179:103104, 2023. 2, 3

[2] R. Ball and C. North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI'05 extended abstracts on Human factors in computing systems*, pp. 1196–1199, 2005. 2

[3] M. Beaudouin-Lafon. Novel interaction techniques for overlapping windows. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 153–154, 2001. 2

[4] V. Biener, S. Kalamkar, N. Nouri, E. Ofek, M. Pahud, J. J. Dudley et al. Quantifying the effects of working in vr for one week. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3810–3820, 2022. 2, 3

[5] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. O. Kristensson et al. Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers. *IEEE transactions on visualization and computer graphics*, 26(12):3490–3502, 2020. 2, 3

[6] V. Biener, F. J. Winston, D. Schmalstieg, and A. Plopski. Long-term experiences from working with extended reality in the wild. *arXiv preprint arXiv:2509.05067*, 2025. 2, 3

[7] S. A. Bly and J. K. Rosenberg. A comparison of tiled and overlapping windows. *ACM SIGCHI Bulletin*, 17(4):101–106, 1986. 4

[8] A. Caetano, A. Aponte, and M. Sra. A design toolkit for task support with mixed reality and artificial intelligence. *Frontiers in Virtual Reality*, 6:1536393, 2025. 3

[9] Y. Cheng, Y. Yan, X. Yi, Y. Shi, and D. Lindlbauer. Semanticadapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 282–297, 2021. 2, 3

[10] Y. F. Cheng, C. Gebhardt, and C. Holz. Interactionadapt: Interaction-driven workspace adaptation for situated virtual reality environments. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, art. no. 109, 14 pp. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3586183.3606717 2, 3

[11] L. A. Cherep, A. F. Lim, J. W. Kelly, D. Acharya, A. Velasco, E. Bustamante et al. Spatial cognitive implications of teleporting through virtual environments. *Journal of Experimental Psychology: Applied*, 26(3):480, 2020. 4

[12] L. Cordioli and M. Matera. From navigation to intention: reframing the web experience through goal-driven interfaces. In *Proceedings of the ACM Web Conference 2026*, pp. 1651–1661, 2026. 3

[13] S. Davari and D. A. Bowman. Towards context-aware adaptation in extended reality: A design space for xr interfaces and an adaptive placement strategy. *arXiv preprint arXiv:2411.02607*, 2024. 3

[14] S. Davari, F. Lu, and D. A. Bowman. Occlusion management techniques for everyday glanceable ar interfaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 324–330. IEEE, 2020. 3

[15] B. Ens, J. D. Hincapié-Ramos, and P. Irani. Ethereal planes: a design framework for 2d information space in 3d mixed reality environments. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pp. 2–12, 2014. 2, 3

[16] B. Ens, E. Ofek, N. Bruce, and P. Irani. Spatial constancy of surface-embedded layouts across multiple environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pp. 65–68, 2015. 2

[17] B. M. Ens, R. Finnegan, and P. P. Irani. The personal cockpit: a spatial interface for effective task switching on head-worn displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3171–3180, 2014. 2, 3, 4

[18] J. a. M. Evangelista Belo, A. M. Feit, T. Feuchtner, and K. Grønbæk. Xrgonomics: Facilitating the creation of ergonomic 3d interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, art. no. 290, 11 pp. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445349 3

[19] J. M. Evangelista Belo, M. N. Lystbæk, A. M. Feit, K. Pfeuffer, P. Kán, A. Oulasvirta et al. Auit–the adaptive user interfaces toolkit for designing xr applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–16, 2022. 2, 3

[20] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2d windows for 3d augmented reality. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pp. 145–155, 1993. 2

[21] A. Fender, P. Herholz, M. Alexa, and J. Müller. Optispace: Automated placement of interactive 3d projection mapping content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2018. 3

[22] A. Fender, D. Lindlbauer, P. Herholz, M. Alexa, and J. Müller. Heatspace: Automatic placement of displays by empirical analysis of user behavior. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 611–621, 2017. 3

[23] W. Feng, W. Zhu, T.-J. Fu, V. Jampani, A. Akula, X. He et al. Layoutgpt: Compositional visual planning and generation with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds., *Advances in Neural Information Processing Systems*, vol. 36, pp. 18225–18250. Curran Associates, Inc., 2023. 3

[24] K. Gajos, D. Christianson, R. Hoffmann, T. Shaked, K. Henning, J. J. Long et al. Fast and robust interface generation for ubiquitous applications. In *International Conference on Ubiquitous Computing*, pp. 37–55. Springer, 2005. 3

[25] C. Gebhardt, B. Hecox, B. van Opheusden, D. Wigdor, J. Hillis, O. Hilliges et al. Learning cooperative personalized policies from gaze data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, pp. 197–208. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347933 3

[26] T. M. Green, W. Ribarsky, and B. Fisher. Building and applying a human cognition model for visual analytics. *Information visualization*, 8(1):1–13, 2009. 2

[27] J. Hartmann, C. Holz, E. Ofek, and A. D. Wilson. Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300577 2, 3

[28] D. Hayatpur, S. Heo, H. Xia, W. Stuerzlinger, and D. Wigdor. Plane, ray, and point: Enabling precise spatial manipulations with shape constraints. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pp. 1185–1195, 2019. 2, 3

[29] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 159–166, 1999. 3

[30] C. A. Johns, J. a. M. Evangelista Belo, C. N. Klokmose, and K. Pfeuffer. Pareto optimal layouts for adaptive mixed reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, art. no. 223, 7 pp. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544549.3585732 3

[31] J. M. Juliano, N. Schweighofer, and S.-L. Liew. Increased cognitive load in immersive virtual reality during visuomotor adaptation is associated with decreased long-term retention and context transfer. *Journal of NeuroEngineering and Rehabilitation*, 19(1):106, 2022. 3

[32] G. Lee, M. Xia, N. Numan, X. Qian, D. Li, Y. Chen et al. Sensible agent: A framework for unobtrusive interaction with proactive ar agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, art. no. 119, 22 pp. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3746059.3747748 3

[33] J. Lee, A. Olwal, H. Ishii, and C. Boulanger. Spacetop: integrating 2d and spatial 3d interactions in a see-through desktop environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 189–192, 2013. 2

[34] Z. Li, J. Chan, J. Walton, H. Benko, D. Wigdor, and M. Glueck. Armstrong: An empirical examination of pointing at non-dominant arm-anchored uis in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2021. 2

[35] Z. Li, C. Gebhardt, Y. Inglin, N. Steck, P. Streli, and C. Holz. Situationadapt: Contextual ui optimization in mixed reality with situation awareness via llm reasoning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, art. no. 43, 13 pp. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3654777.3676470 2, 3

[36] J. Lin, J. Guo, S. Sun, Z. J. Yang, J.-G. Lou, and D. Zhang. Layoutprompter: Awaken the design ability of large language models, 2023. 3

[37] D. Lindlbauer, A. M. Feit, and O. Hilliges. Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd Annual ACM

Symposium on User Interface Software and Technology, UIST '19, pp. 147–160. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347945 3

[38] T. Liu, W. He, and M. Billinghurst. Using virtual reality as a simulation tool for augmented reality virtual windows: Effects on cognitive workload and task performance. *arXiv preprint arXiv:2409.16037*, 2024. 3

[39] F. Lu and Y. Xu. Exploring spatial ui transition mechanisms with head-worn augmented reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, art. no. 550, 16 pp. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491102.3517723 3

[40] Y. Lu, Z. Tong, Q. Zhao, C. Zhang, and T. J.-J. Li. Ui layout generation with llms guided by ui grammar. *arXiv preprint arXiv:2310.15455*, 2023. 3

[41] Y. Lu, Z. Tong, Q. Zhao, C. Zhang, and T. J.-J. Li. Ui layout generation with llms guided by ui grammar, 2023. 3

[42] W. Luo, M. O. Ellenberg, M. Satkowski, and R. Dachselt. Documents in your hands: Exploring interaction techniques for spatial arrangement of augmented reality documents. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2025. 2

[43] M. N. Lystbæk, K. Pfeuffer, T. Langlotz, J. E. S. Grønbæk, and H. Gellersen. Spatial gaze markers: Supporting effective task switching in augmented reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2024. 3

[44] Microsoft. Typography – mixed reality. Microsoft Learn, 2021. Accessed: 2026-07-22. 5

[45] A. Niyazov, B. Ens, K. A. Satriadi, N. Mellado, L. Barthe, T. Dwyer et al. User-driven constraints for layout optimisation in augmented reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, art. no. 35, 16 pp. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580873 2, 3

[46] L. Pavanatto, J. Grubert, and D. A. Bowman. Spatial bar: Exploring window switching techniques for large virtual displays. In *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 186–194. IEEE, 2025. 2, 3

[47] L. Pavanatto, F. Lu, C. North, and D. A. Bowman. Multiple monitors or single canvas? evaluating window management and layout strategies on virtual displays. *IEEE Transactions on Visualization and Computer Graphics*, 31(3):1713–1730, 2024. 1, 2

[48] L. Pavanatto, C. North, D. A. Bowman, C. Badea, and R. Stoakley. Do we still need physical monitors? an evaluation of the usability of ar virtual monitors for productivity work. In *2021 IEEE virtual reality and 3D user interfaces (VR)*, pp. 759–767. IEEE, 2021. 1, 2, 3

[49] S. Pei, D. Kim, A. Olwal, Y. Zhang, and R. Du. Ui mobility control in xr: Switching ui positionings between static, dynamic, and self entities. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2024. 2, 3

[50] L. Pietschmann, M. Schimpf, Z.-T. Chen, H. Pfister, and T. Bohné. Enhancing user performance and human factors through visual guidance in ar assembly tasks. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2025. 2

[51] J. Ren, Y. Weng, C. Zhou, C. Yu, and Y. Shi. Understanding window management interactions in ar headset+ smartphone interface. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2020. 2

[52] G. Robertson, E. Horvitz, M. Czerwinski, P. Baudisch, D. R. Hutchings, B. Meyers et al. Scalable fabric: flexible task management. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 85–89, 2004. 2

[53] G. Robertson, M. Van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden et al. The task gallery: a 3d window manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 494–501, 2000. 2, 3

[54] A. Ruvimova, J. Kim, T. Fritz, M. Hancock, and D. C. Shepherd. " transport me away": Fostering flow in open offices through virtual reality. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–14, 2020. 2, 3

[55] D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pp. 20–29. IEEE, 2000. 1

[56] M. Serrano, B. M. Ens, and P. P. Irani. Exploring the use of hand-to-face input for interacting with head-worn displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3181–3190, 2014. 2, 3

[57] J. N. Setu, J. M. Le, R. K. Kundu, B. Giesbrecht, T. Höllerer, K. A. Hoque et al. Predicting and explaining cognitive load, attention, and working memory in virtual multitasking. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 3

[58] Y. Song, C. Gebhardt, Y.-C. Liao, and C. Holz. Preference-guided multi-objective ui adaptation. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, art. no. 120, 13 pp. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3746059.3747645 2, 3

[59] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–272, 1995. 3

[60] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock. Mixed reality remote collaboration combining 360 video and 3d reconstruction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–14, 2019. 2

[61] H. Vijay, S. Pushp, A. Mittal, P. Gupta, M. Gupta, S. Gambhira et al. Hyway: Enabling mingling in the hybrid world. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(2):1–33, 2023. 3

[62] J.-D. Wang, K. Zhou, H. Ren, P. O. Kristensson, and X. Li. Handows: A palm-based interactive multi-window management system in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2, 3

[63] D. Wolf, J. Gugenheimer, M. Combosch, and E. Rukzio. Understanding the heisenberg effect of spatial interaction: A selection induced error for spatially tracked input devices. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–10, 2020. 2

[64] Y. Yan, C. Yu, X. Ma, S. Huang, H. Iqbal, and Y. Shi. Eyes-free target acquisition in interaction space around the body for virtual reality. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pp. 1–13, 2018. 2

[65] L. Zhang, J. Pan, J. Gettig, S. Oney, and A. Guo. Vrcopilot: authoring 3d layouts with generative ai models in vr. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–13, 2024. 3

[66] H. Zhou, T. Ayesh, C. Fan, Z. Sarsenbayeva, and A. Withana. Coplayingvr: Understanding user experience in shared control in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(3):1–25, 2024. 3

[67] H. Zhou, T. Kip, Y. Dong, A. Bianchi, Z. Sarsenbayeva, and A. Withana. Juggling extra limbs: Identifying control strategies for supernumerary multi-arms in virtual reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2025. 3

[68] Q. Zhou, G. Fitzmaurice, and F. Anderson. In-depth mouse: Integrating desktop mouse into virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2022. 2, 3

[69] C. Zhu, S.-K. Hsia, X. Hu, Z. Liu, J. Shi, and K. Ramani. agentar: Creating augmented reality applications with tool-augmented llm-based autonomous agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST '25, art. no. 54, 23 pp. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3746059.3747676 3