

Evaluating Navigation and Comparison Performance of Computational Notebooks on Desktop and in Virtual Reality

Sungwon In
Virginia Tech
Blacksburg, United States
sungwoni@vt.edu

Eric Krokos
Department of Defense
United States
ericpkrokos@gmail.com

Kirsten Whitley
Department of Defense
United States
visual.tycho@gmail.com

Chris North
Virginia Tech
Blacksburg, United States
north@cs.vt.edu

Yalong Yang
Georgia Tech
Atlanta, United States
yalong.yang@gatech.edu



Figure 1: A demonstration of the immersive computational notebook system featuring the branching capabilities. The user can work with windows containing multiple cells, which are interconnected to indicate execution order. (a) and (b) shows branches created at different locations of the notebook for testing multi-level hypotheses. (c) subsequently merge these branches for streamlined result comparisons, with six results—three branches at (a) \times two branches at (b)—presented in the output of the window.

ABSTRACT

The computational notebook serves as a versatile tool for data analysis. However, its conventional user interface falls short of keeping pace with the ever-growing data-related tasks, signaling the need for novel approaches. With the rapid development of interaction

techniques and computing environments, there is a growing interest in integrating emerging technologies in data-driven workflows. Virtual reality, in particular, has demonstrated its potential in interactive data visualizations. In this work, we aimed to experiment with adapting computational notebooks into VR and verify the potential benefits VR can bring. We focus on the navigation and comparison aspects as they are primitive components in analysts' workflow. To further improve comparison, we have designed and implemented a Branching&Merging functionality. We tested computational notebooks on the desktop and in VR, both with and without the added Branching&Merging capability. We found VR significantly facilitated navigation compared to desktop, and the ability to create branches enhanced comparison.



This work is licensed under a Creative Commons Attribution International 4.0 License.

CCS CONCEPTS

- Human-centered computing → Empirical studies in HCI; Interactive systems and tools; Virtual reality.

KEYWORDS

immersive analytics, computational notebook system, data science, 3D UI & interaction

ACM Reference Format:

Sungwon In, Eric Krokos, Kirsten Whitley, Chris North, and Yalong Yang. 2024. Evaluating Navigation and Comparison Performance of Computational Notebooks on Desktop and in Virtual Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3613904.3642932>

1 INTRODUCTION

The computational notebook has gained substantial popularity across diverse domains due to its versatility, characterized by its seamless integration of code, documentation, and visual outputs in a unified interface, ease of sharing and replication, and interactive features with real-time feedback. Data analysts leverage these capabilities for tasks ranging from constructing analytical pipelines to debugging and comparative analysis [9].

However, as data analysis grows in complexity, the standard computational notebook's user interface shows limitations in supporting the aforementioned tasks. Specifically, *navigating* through extensive notebooks becomes increasingly challenging, complicating tasks like identifying issues in the analytical pipeline or refactoring code [31]. Analysts often have to scroll up and down through lengthy sections multiple times, relying heavily on their working memory for off-screen content, leading to high context-switching costs [30]. Additionally, modern data analysis often involves *comparisons*, such as determining optimal parameters for analytical models. Common practices, like duplicating code or notebook, complicate code and execution management by introducing non-linearities [15]. While literature identifies other challenges with computational notebooks [9], this project focuses on enhancing *navigation* and *comparison*, which are primitive components of data analysts' workflows [55].

Efforts to improve navigation and comparison in computational notebooks within the desktop environment exist [68]. However, the inherent limitations of desktop display and interaction paradigms restrict the spatial presentation and interaction modes with computational notebooks. Thus, we explore opportunities offered by emerging technologies, specifically virtual reality (VR). VR headsets allow interaction with 2D and 3D graphics and interfaces in an expansive space, introducing new human-computer interaction possibilities for non-linear notebooks. Preliminary research in using VR for data science shows promising benefits. For example, VR allows analysts to use its large space as an external memory layer with spatial semantic meanings to better support information retrieval [14, 40, 59]. Physical interactions in VR, including natural walking and embodied gestures, provide rapid information access and command execution [25, 26, 65, 70]. Consequently, our overarching research question is: **Can VR's spatial and embodied**

nature enhance navigation and comparison in computational notebooks?

To approach the proposed research question, we first adapted the computational notebook for the VR environment. Specifically, we introduced an additional hierarchy layer to facilitate notebook content management, adopted a curved layout for content placement, and designed gesture-based interactions, including a branch&merge gesture to assist comparison. To systematically verify and understand the potential benefits of virtual reality, we conducted a controlled user study to compare computational notebooks on the desktop and in VR, both with and without branch&merge capability. The study task comprised two phases: first, participants were asked to *navigate* through a presented computational notebook, identifying and rectifying deliberate issues; second, they were required to determine optimal parameter values through *comparisons*. Pre-configured codes were provided to reduce the need for constructing a notebook from scratch, enabling participants to focus on *navigation* and *comparison* tasks. We noticed that participants encountered significant challenges in editing text while in VR. After excluding the text editing time from all conditions, we found that VR had better navigation performance than the desktop, and the branch&merge functionality significantly facilitated the comparison process. The contributions of this work are twofold: 1) the adaptation of computational notebooks from desktop to VR, and 2) empirical knowledge about using computational notebooks in VR.

2 RELATED WORK

2.1 Challenges in Computational Notebooks

Drawing from Knuth's literate programming paradigm [34], computational notebooks seek to construct a computational narrative, enhancing analysts' efficiency in iterative data science tasks by amalgamating visuals, text, and analytical insights into an integrated document [60]. Numerous implementations, such as Jupyter Notebook [28], DataBricks [12], Apache Zeppelin [77], and CarbidAlpha [8], have been developed. Yet, as data analyses evolve in complexity, these platforms present challenges in supporting the increasingly intricate data science workflows. In 2015 and 2020, Jupyter, a leading computational notebook application, conducted user surveys to illuminate these issues [27], receiving feedback mainly on system functionalities like encompassing performance, sharing capabilities, version control, and enriched documentation. Certain user experience concerns also emerged, like content collapse, progress indicators, and global search. Aligning with this endeavor, Chattopadhyay et al. [9] undertook a rigorous exploration involving 156 data science professionals to systematically unravel the pain points, needs, and opportunities with computational notebooks. Their study spotlighted nine pivotal challenges that not only add operational hurdles but also layer on complexities detrimental to analytical workflows.

Our study particularly addresses challenges tied to the visual and interactive facets of computational notebooks, with an emphasis on exploration and code management. As underscored by previous research [24, 60], during exploratory phases, analysts often prioritize flexibility and speed over clarity and sustainability, leading to long and "messy" notebooks. A prevalent practice includes cloning variables, code segments, or entire notebooks as an informal versioning

method, bypassing standard tools [31, 64]. As a result, unintentional modifications and deletions in notebooks make data analysis error-prone and laborious [31, 60], and locating specific elements becomes more challenging with increasing complexity, as echoed in the Jupyter survey [31, 60, 64]. Taking cues from prior research, we focus on supporting *navigation* and *comparison* of computation notebooks by leveraging the display and interaction capabilities of immersive environments.

2.2 Navigation and Comparison in Computational Notebooks

Improving the navigation experience has historically been a central focus in user interface and interaction design, resulting in various techniques for diverse applications. Among them, notably, techniques like focus+context and overview+detail have gained significant traction. Cockburn et al. [10] provided an exhaustive review of these techniques. Focus+context techniques like the fisheye lens in image viewing are suitable for certain applications [18, 57, 74], but their distortion effects and extensive adaptation requirements limit their suitability in text-dense environments like computational notebooks. Overview+detail design, featuring a separate overview window that displays a thumbnail of the entire content, aids users in identifying their position and finding specific sections. This approach is integrated into some text editors, IDEs, and even computational notebooks like Google Colab [20], where it presents an “outline” view based on the markdown hierarchies. Nevertheless, users require extra and explicit effort to create and maintain the hierarchies or summaries. In this research, we hypothesize that the inherent spatial and physical navigation offered by immersive environments can streamline notebook navigation.

Analyzing different hypotheses often mandates the development of multiple versions of analyses or implementations, followed by result comparisons. Managing these versions poses challenges and is often error-prone. Hartmann et al. [22] developed interfaces presenting results from various alternatives in a unified view to simplify comparison. Weinman et al. [68] adopted these concepts to computational notebooks, proposing the ability to create multiple non-linear execution paths. Here, users can “fork” content from a chosen cell, but the approach’s limitation to full code duplication and lack of support for simultaneous paths may restrict its use and increase maintenance. As an alternative, Harden et al. introduced a 2D computation notebook, giving users the flexibility to organize cells and results bidimensionally, facilitating easier side-by-side comparisons [21]. Inspired by these precedents, we aim to empower analysts to replicate only essential content for creating comparisons and to position comparison results adjacently for efficiency.

2.3 Immersive Analytics

Immersive analytics represents an emerging research field exploring the integration of novel interaction and display technologies to enhance data analytics, particularly through the lens of VR/AR [19, 45, 78]. Current studies in immersive analytics predominantly emphasize the data visualization aspect and have identified several key benefits of using VR/AR. For instance, previous work reported rendering 3D network graphs in VR to be more effective than on

flat screens due to the added dimension to declutter the visual information [11, 37, 71]. The large display space in VR also permits users to organize content spatially [23, 38, 62], enhancing physical navigation such as walking and head movement, found to be more effective than virtual methods like pan&zoom [5]. Research further highlights spatial memory plays a crucial role in VR/AR information retrieval [69], and VR/AR’s accurate motion tracking offers opportunities for intuitive interaction designs, like the Tilt Map [72] that enables switching between 2D and 3D visualizations. It’s worth noting that the cited examples merely offer a snapshot of the myriad VR/AR advantages documented in literature rather than an exhaustive list.

Building on empirical evidence that attests to the advantages of VR/AR in data visualization, researchers in the field of immersive analytics have begun to explore whether these benefits can be extended to the broader scope of data analytics. For instance, In et al. [26] developed a tool that facilitates gesture-based interactive data transformation within a VR environment. In a similar vein, Lisle and Davidson et al. [14, 39, 40] introduced the concept of an “Immersive Space to Think,” leveraging the expansive display capabilities of VR/AR for improved text content management and insight generation. Luo et al. [44] examined strategies for spatially organizing documents in AR settings. In alignment with these pioneering efforts, our research aims to explore the potential benefits that immersive environments could offer to computational notebook applications.

3 ADAPTING COMPUTATIONAL NOTEBOOKS TO VR

Our primary objective is to examine the user experience of computational notebooks in VR and to explore the potential advantages VR may offer. A crucial initial step is adapting the computational notebook system for VR use. While the desktop version of the computational notebook is well-established, transitioning it to a VR setting introduces unique challenges. These include determining how to visually represent and spatially position the notebook in the VR space and identifying necessary interactions to facilitate its use in this immersive environment. As a preliminary step for designing VR-compatible computational notebooks, we aimed for a smooth transition for analysts by maintaining design consistency with familiar desktop counterparts, while also leveraging VR’s distinctive capabilities where beneficial. This section details our design objectives, centered on enhancing the *navigation* and *comparison* functionalities of computational notebooks, and discusses our primary design considerations and decisions.

3.1 Design Goals

Navigating a computational notebook involves an analyst shifting their focus to a different section of the notebook by actively changing the content displayed in their field of view (FoV). Navigation is fundamental to various tasks performed by analysts using computational notebooks. Analysts frequently navigate between various parts of their analytical pipeline during exploratory data analysis to derive insights from data. Similarly, navigation is essential when taking over someone else’s project to understand their process or when compiling a final report from the analysis conducted [29].

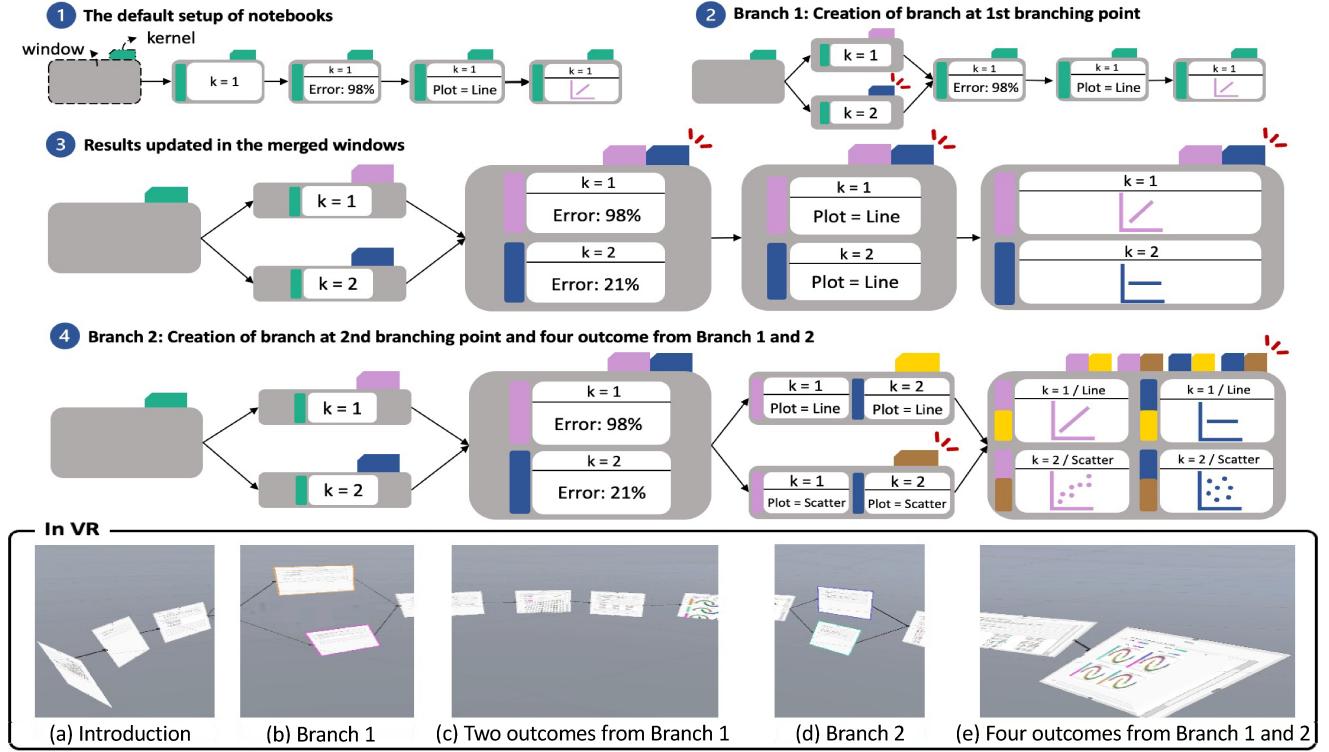


Figure 2: Visual representation (top) of the Branch&Merge mechanism within computational notebooks, as instantiated in a VR (bottom). Different hues within the results are employed to illustrate the independent storage of variables across branches. The figure is segmented into five key operations: (a) regular notebooks, (b) initial creation of the branch, (c) merge back to a singular code execution path, (d) initiation of the second branch, and (e) final merging process to facilitate value comparisons across all initiated branches.

Building upon prior research [6, 49, 53, 54, 70], we define navigation as a multi-component process, primarily involving locating target sections and then moving towards them. With a conventional desktop computational notebook, it is challenging for analysts to recall the location of off-screen targets and accurately navigate (“scroll”) to these locations. Therefore, our goal is to *enable analysts to quickly identify (locate) and reach (move to) their desired sections within the computational notebook in VR*.

Comparing outcomes derived from varying parameters or methods is a frequent task for analysts working with computational notebooks [41]. This process involves two stages: initially establishing the comparisons by coding tests for different parameters or methods, and subsequently examining the results to make informed choices regarding these parameters or methods. In the setting of traditional desktop computational notebooks, analysts typically rely on their memory for comparison tasks, externalize results for comparisons, write specialized code to produce multifaceted results, or replicate the notebook for comparison purposes. However, each of these methods has its limitations, either placing a significant cognitive load on the analyst’s working memory or complicating the management of code and content. Consequently, our objective is to *enable analysts to intuitively generate comparisons and easily review all generated results*.

3.2 Adaptations

To investigate the advantages that VR could offer to computational notebooks, we designed and implemented adaptations that capitalize on VR’s distinctive display and interaction features, with a focus on enhancing the *navigation* and *comparison* experiences. We specifically focused on facilitating physical navigation within the VR environment and enhancing comparison tasks through an embodied branch&merge gesture. Furthermore, we also developed other essential interactions tailored to the notebook’s functionality in VR.

Enabling Physical Navigation. Physical navigation involves utilizing bodily movements, such as head rotation or walking, to explore an information space, like accessing various sections of a computational notebook. Research by Ball et al. [5] demonstrated that this form of navigation is more efficient than virtual navigation (pan and zoom) in the context of browsing geographic maps on large display walls. In VR, the ability to spatially arrange content offers a unique opportunity for physical navigation, which we anticipate could enhance notebook navigation compared to desktop versions. To establish an environment conducive to physical navigation, we implemented the following adaptations.

Adding one hierarchical layer. Desktop computational notebooks employ a linear arrangement of cells and outputs within a singular window. Yet, transplanting this design directly to an immersive

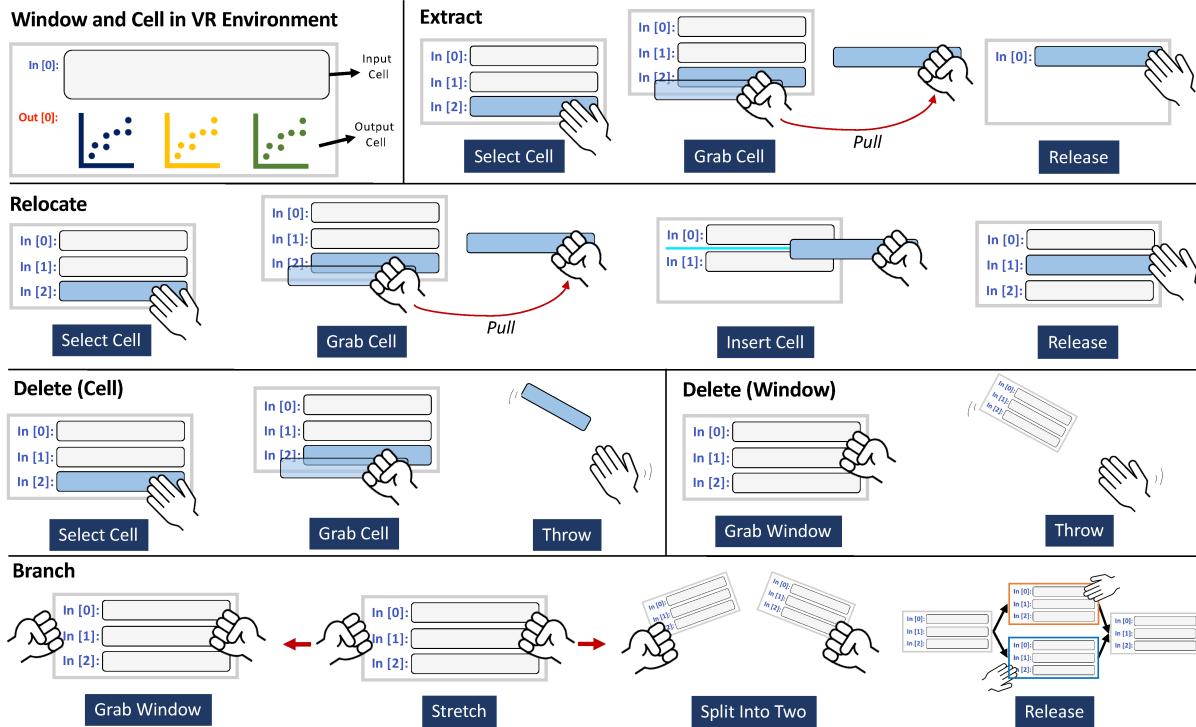


Figure 3: Illustrations of gestures interactions in the VR environment.

environment poses challenges. Extending the single window to accommodate all content could result in an impractically long display, potentially falling outside the user's reach. To convert a notebook into a suitable format for spatial distribution, we introduce an additional hierarchical layer to the content layout: cells and outputs are organized within individual windows, and these windows are interlinked to compose a complete notebook, see Fig. 1.

Applying a curved layout. Drawing on observations from Andrews et al. [2] regarding the layout of multiple windows, we adopted a commonly used horizontal window placement strategy. This linear arrangement, signified by directed arrows from left to right, not only clarifies the sequential order of windows but also ensures that all windows fall within the user's vertical reach. To optimize the curvature of this arrangement, we consulted Liu et al.'s findings [42, 43] for our initial layout placement, which indicate that a semi-circular layout generally surpasses both flat and full-circle configurations.

Embodied Branch&Merge for Comparison. In hypothesis testing via comparisons, analysts frequently employ the strategy of creating additional copies and modifying relevant content, such as adjusting a variable's value or invoking a different function, as highlighted by Weinman et al. [68]. In response, they introduced an interactive tool named “fork it,” enabling users to create a concurrent copy with a button click. Adapting this interactive concept to immersive settings, we designed an embodied gesture for duplication: users grasp the window they wish to replicate with both hands and then stretch it until a specific threshold, as illustrated in Fig. 3—Branch. The user can freely place the newly created windows in space.

Our enhanced hierarchical structure in notebook content organization offers increased flexibility for hypothesis evaluation through non-linear branching. This enables precise content duplication at the window's granularity, as opposed to copying subsequent content, as seen in the “fork it” [68]. For instance, when an analyst intends to probe different predefined cluster values in K-means clustering, they can produce branches for various cluster assignments. The subsequent visualization code used to assess clustering results remains unduplicated. Moreover, our system supports branching at multiple points simultaneously, unlike prior systems that were limited to single-point branching. For instance, consider having a linear notebook without any branch initially, illustrated in Fig. 2-1. Subsequently, the user can create a branch at any point, as demonstrated in Fig. 2-2. The creation of a branch leads to added results for comparisons in all the subsequent windows. The number of results is equal to the number of branched windows, say two, as depicted in Fig. 2-3. Furthermore, the analyst can create another branch as demonstrated in Fig. 2-4, and the subsequent window (only one last window in this case) will have four results produced by all combinations of the two branches.

Additional Notebook Interactions. Computational notebooks come with fundamental interactions for content management, such as creating, deleting, and moving cells, typically one cell at a time. Initially, we explored a toolbar-based design situated at the top of the window, mirroring the traditional desktop-based computational notebook interface. Nevertheless, our internal evaluations highlighted challenges with using a pointer for button interactions, confirming findings from previous studies [47, 61]. A recent study by In et al. highlighted the advantages of gesture-based interactions

over the window-icon-menu-pointer (WIMP) design [26] in immersive settings. Consequently, we pivoted to designing intuitive gesture interactions for the immersive environment, which are also demonstrated in Fig. 3.

- **Extract:** Users can deploy a “grab & pull” gesture to detach a cell from its window, creating a new window for the selected cell. When extracting multiple cells, users can first select them and then employ the “grab & pull” gesture.
- **Delete:** To remove a cell, users can utilize the “grab & throw away” gesture. For multiple cells, after selection, the same gesture is employed. Entire windows can similarly be discarded with this gesture.
- **Relocate:** Users can “grab” a cell and “drag” it to a new position, whether within its initial window or to a different one. To relocate multiple cells, users should first select them and then execute the “grab & drag” action.

Additional interactions include “grab & move” for repositioning windows and “pinch-to-zoom” for resizing, where users collide their hands with a window and move them apart to enlarge or together to minimize. Beyond content and view adjustments, a single button on the user’s left hand, inspired by Yang et al. [70], proved more efficient than attaching buttons to each window. This “Run” button executes the selected cell and all subsequent cells.

4 USER STUDY AND EVALUATION

To systematically answer the research question, we designed and conducted a controlled user study. Primarily, our objective was to explore the potential advantages of utilizing an immersive environment. To this end, we compared our immersive computational notebook implementation with its DESKTOP counterpart. Additionally, we sought to enrich empirical evidence supporting the merits of the “branch&merge.” In summary, our study encompassed four conditions: DESKTOP+LINEAR, DESKTOP+BRANCH, VR+LINEAR, and VR+BRANCH. These conditions enabled a systematic investigation of the effect of two variables: the computing environments (□ Desktop vs. ⚡ VR) and the comparison techniques (Linear vs. Branch).

4.1 Study Conditions

Our ⚡ Branch design in the immersive computational notebook (VR+BRANCH) is presented at Sec. 3—Branch&Merge. Meanwhile, our DESKTOP+BRANCH implementation shared a similar idea but utilized a button to duplicate a window instead of using an embodied gesture. On the other hand, the creation of branches was not permitted in both the DESKTOP+LINEAR and VR+LINEAR conditions. In the following, we detail our □ Desktop and ⚡ VR implementations.

□ Desktop: In the DESKTOP environment, our design emulates the features and functionalities typical of standard computational notebooks, where interactions are facilitated via a mouse and keyboard. To navigate different segments of the notebook, we incorporated vertical scrolling, a standard navigation method in many DESKTOP applications. Users can navigate by using the mouse scroll or dragging the scrollbar. For other content interactions, we also follow the standard computational notebook designs: users use the mouse to select the target and click the buttons to execute specific

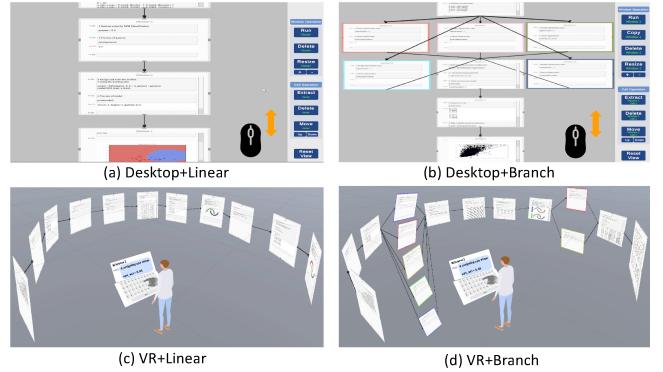


Figure 4: Demonstrations of conditions tested in our study. User scrolling to navigate in DESKTOP (top), and walking and rotating their body to navigate in VR (bottom). In addition, the user used a physical keyboard to write codes in DESKTOP, while used a virtual keyboard in VR.

commands, like extract, delete, and relocate. To ensure an equitable comparison and to control for potential confounders, we integrated features detailed in Sec. 3.

⚡ VR: We detail our primary computational notebook designs and implementations for VR in Sec. 3. This section focuses on vital design choices not strictly tied to computational notebook features. We opted for bare-hand interaction over using controllers, aiming for a more intuitive and immersive user experience.

However, text input remains a crucial aspect of immersive computational notebooks. Meta has pioneered a technology that brings physical keyboard tracking into VR [46], which we initially adopted. However, Meta’s design primarily suits a seated work environment, as carrying and typing on a physical keyboard while navigating in VR is impractical. While Davidson et al. [14] proposed using a mobile table for the physical keyboard, their approach inadvertently tethered users to the table, reducing spatial exploration. To encourage fuller utilization of physical navigation in VR, we decided not to adopt their method, opting for the standard virtual keyboard method instead. Our first virtual keyboard implementation employed an “on-demand” approach, appearing during text interaction and vanishing when interacting with other elements. However, this approach was changed due to internal tests showing it often activated by mistake, leading to cluttering the interface. The revised design, inspired by Yang et al. [70], attaches the keyboard to the left palm, with improvements to prevent unintentional activations by requiring users to intentionally look at their palm to activate it.

In VR settings, interacting with distant windows presents a challenge for various text interactions. We selected a font size that ensured text readability from the initial viewing distance, eliminating the need for users to continuously adjust their proximity for code visibility. While the text remains readable from afar, pinpointing and selecting a precise point within the text, such as a specific entry, remains challenging. To facilitate precise interaction, we implemented a design inspired by Voodoo dolls [52], where a proximate copy of the interacting cell is generated for the user. Interactions between this close copy and the original notebook window are synchronized, as illustrated in Fig. 4 (c) and (d).

4.2 Task and Data

Computational notebooks are versatile tools that facilitate a wide range of analytical tasks, from creating new notebooks to utilizing existing code for data analysis. However, requiring participants to extensively write code from scratch in a controlled study could be time-consuming and introduce confounding factors outside the scope of our investigation. Meanwhile, it's increasingly common for analysts to revisit or leverage existing code, whether it's from inheriting someone else's project or reusing their previous work [1]. To maintain the focus of our study on the aspects of navigation and comparison, we simulated a scenario where participants were provided with pre-existing code.

We structured two tasks to evaluate the navigation and comparison performance within our test conditions. To mitigate any learning effects, each participant was presented with four analytical methods, assigning one method to each test condition: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Linear Regression. As mentioned, the order was determined by a balanced Latin square matrix. The computational notebooks used in the study were sourced from the official scikit-learn documentation [51]. These notebooks were standardized to approximately ten windows each, based on internal testing that showed our chosen length allowed for practical task completion times. Moreover, the notebooks exhibited a consistent logical structure: they initiated with data loading, data transformation, modeling, and concluding with the visualization of model outcomes. Details of the specific study tasks, formulated within this overarching context, are presented subsequently. Supplementary materials containing all study stimuli are provided.

Task 1. Navigation. To assess the performance across different types of navigation, we designed tasks that involved both single-stop and multi-stop (specifically, two-stop) navigation scenarios. The single-stop task had participants identify and delete an error-causing cell called *deletion*, while the two-stop task involved identifying and correctly repositioning a misplaced cell called *relocation*. Participants could identify these target cells by inspecting the cells' output; notably, cells generating errors and their subsequent cells would not yield output. This task was designed to be navigation-intensive, enabling us to extract nuanced differences in performance and user experience related to navigation activities. Consequently, our primary aim was to investigate the impact of computational environments—VR vs. DESKTOP—on navigation efficiency and user experience.

Task 2. Comparison. Following Task 1, we introduced Task 2, designed to simulate real-world hypothesis testing. Participants evaluated two parameters within various analytical methods, such as cluster numbers and distance metrics in the KNN method, to determine the optimal parameter combinations based on the visualized results. To ensure consistent experimental conditions, we standardized the spatial distance of relevant windows (i.e., the two windows containing “what-if” tests and the result window) across all trials. Minimal text input was required from participants due to pre-commented code, simplifying the process of parameter adjustment. Although our primary focus was on comparison, navigation was inherently involved in task completion.

4.3 Experimental Setup

In the VR configuration, we utilized the Meta Quest Pro headset, providing a resolution of 1800×1920 . For DESKTOP conditions, a standard 27" monitor was deployed, featuring a 2560×1440 resolution. The Meta Air Link feature was used for the VR environments, enabling a tether-free experience by leveraging the PC for computations while the headset managed to render. This configuration allowed participants to freely navigate the $16m^2$ space without worrying about cable impediments. Within the VR setting, participants started at the center of the space, presented with ten notebook windows, each measuring $0.35 \times 0.30m^2$, arranged semi-circularly at a distance of 1m. Conversely, DESKTOP participants sat at a desk, encountering initial notebook windows followed a linear structure and sized at on average 2000×600 pixels each. Specifically, the notebook windows were displayed on the center of the monitor, allowing participants to view two or three notebook windows at the same time. Additionally, for the DESKTOP+BRANCH, we left empty spaces on both the left and right sides of the notebook windows to ensure sufficient space for branching. This arrangement was mirrored in the DESKTOP+LINEAR to ensure consistency across the study settings, as shown in Fig. 4 (a) and (b).

4.4 Participants

We recruited 20 participants (16 male, four female, ages 18 to 35) from a university mailing list. The recruitment was based on their knowledge of data science and machine learning algorithms, which was screened using an eight-question quiz (provided in the supplementary material). Participants need to answer six out of eight to be eligible for the study. Out of the 22 respondents, 20 were invited to participate in the study based on the eligibility requirement. Regarding VR experience, seven of the participants use VR weekly, and the remaining thirteen have no prior VR experience. All participants had either normal vision or vision corrected to normal. For their time and contribution to the study, each participant was compensated with a \$20 Amazon Gift Card.

4.5 Design and Procedures

Our user study followed a full-factorial within-subjects design, with conditions balanced using a Latin square (4 groups). The study, on average, took less than two hours. Participants were initially welcomed and reviewed a consent form. Then, we briefly introduced the study's objectives and procedural steps. Following this introduction, participants proceeded to the various components of the study as follows:

Preparation: We asked participants to adjust the chair height to a comfortable level for the DESKTOP condition and adjust the Quest Pro headset for the VR condition before they started the training session. We confirmed that all participants were in comfortable conditions and could see the text in all display environments clearly.

Training: We initiated our study by standardizing computational notebook terminologies, recognizing the potential for varied interpretations. The training was provided only when participants first encountered a computing environment (i.e., Desktop or VR). This was due to the consistent operational logic within each environment, with differences only in the comparison task. In the training, participants viewed operational demonstration videos.

Post-viewing, we verified their understanding, asking them to replicate study tasks using a different algorithm, k-mean. Participants were free to inquire about operations or tasks. The training sessions, particularly for the VR condition, were extended to give participants enough time to become comfortable with the immersive environment. This approach was adopted to address potential VR-related issues such as discomfort, learning effects, and novelty bias. In summary, the training was completed once participants achieved proficiency in tasks and operations, which generally took 10-15 minutes.

Study Task: Upon completion of the training session, participants proceeded to the study task. To ensure they had enough understanding of what would be expected, we provided comprehensive context, including a brief explanation of the algorithms and the tasks they needed to complete. Participants had no time limit for task completion but were encouraged to prioritize accuracy and efficiency. For the VR environment, we reset the participants' position to the center of the room and had them face the same initial direction before each study task started.

Questionnaires. *Post-Condition Questionnaires:* upon completion of each condition, participants were required to fill out a Likert-scale survey. This was adapted from the System Usability Scale (SUS) and NASA Task Load Index (TLX) to record their subjective experiences. Additionally, they were asked to provide qualitative feedback concerning the pros and cons of the condition they had just interacted with. *Post-Study Questionnaires:* Once all the study tasks and post-condition questionnaires were completed, participants were asked to rank the study conditions based on their overall experience.

4.6 Hypotheses

We aimed to validate whether our designs and implementations met our established design goals. Consequently, we formulated hypotheses grounded in empirical findings from prior research and the testing conditions described in Sec. 4.1.

Navigation in VR and on DESKTOP (H_{nav}). We hypothesized VR would provide faster navigation than DESKTOP. VR offers a large display space to lay out an entire notebook, allowing participants to navigate by physically walking in the space or rotating their heads. In contrast, DESKTOP presents only part of a notebook at a time, requiring participants to scroll up and down for navigation. Previous studies indicate that physical navigation—employed in our VR conditions—is more effective than virtual navigation, as used in DESKTOP conditions [5]. Furthermore, VR has been demonstrated to enhance spatial awareness, thereby aiding in the recall process during multi-stop navigations [36, 69]. We believe these established advantages of VR are generalizable to computational notebooks.

Comparison with and without BRANCH ($H_{comp-branch}$). We expected that the incorporation of the BRANCH feature would facilitate comparison tasks. Earlier studies have reported favorable user experiences with similar functionalities in computational notebooks [21, 68]. Building on these insights, we introduced additional features, such as merging post-branching, to minimize visual clutter and spatially organize results. We aim to provide quantitative empirical data to highlight the effectiveness of the BRANCH functionality.

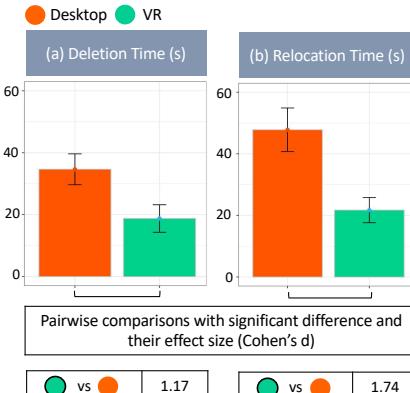


Figure 5: Completion time for VR and DESKTOP in the navigation task. (a) the time spent for completing deletion, and (b) the time spent completing relocation. Solid lines indicate statistical significance with $p < 0.05$. The tables below show the Cohen's D effect sizes for significant comparisons. Circles with black borders indicate the condition with better results.

Comparison in VR and on DESKTOP ($H_{comp-env}$). We anticipated that VR could outperform DESKTOP in performing comparisons. In the LINEAR conditions of DESKTOP and VR, the comparison would be intrinsically linked to the navigation efficiency, as participants would only view one result at a time in both conditions. Given this, the navigational advantages of VR are expected to positively impact comparison tasks in LINEAR conditions. For the BRANCH conditions, we considered our designed VR embodied gesture for branch creation would be intuitive, thereby facilitating the process. Moreover, the large display space in VR could allow participants to view all results simultaneously, expediting the visual assessment process.

4.7 Measures

In our study, we gathered quantitative data to evaluate our hypotheses. For the navigation task, we logged the *time* taken by participants to complete one-stop (deletion) and two-stop (relocation) navigations under each condition. *Completion times* for the comparison task were also recorded, as was the frequency of “Run” button presses, indicating execution. For potential subsequent analyses, we also logged user interaction and tracked objects in the scene.

After participants engaged with a specific condition, they were asked to complete a survey using a 7-point Likert scale to gauge their perceived *physical* and *mental demands*, *engagement*, and the *effectiveness* of that condition. To further explore the nuances of each condition, semi-structured interviews were conducted, highlighting both strengths and areas for improvement for each condition. Concluding the study, participants provided an overall ranking of their user experience within the testing conditions.

5 RESULTS

In this section, we present the statistical analysis of our collected data, outline the strategies participants employed to manage the display space, and provide summarized qualitative feedback for each condition. We documented significance at levels of $p < 0.001$ (***)



Figure 6: Analysis of four testing conditions in the comparison task. (a) the time spent completing the task, (b) the number of executions performed, (c) the time spent for text input, (d) the time spent completing the task excluding the text input interaction time, (e) the total time spent for creating all branches in DESKTOP and VR, (f) the average time spent for creating a branch in DESKTOP and VR. Solid lines indicate statistical significance with $p < 0.05$, and dashed lines indicate $0.05 < p < 0.1$. The tables below show the Cohen's D effect sizes for significant comparisons. Circles with black borders indicate the condition with better results.

$p < 0.01^{(\star\star)}$, $p < 0.05^{(\star)}$, and $p < 0.1^{(\cdot)}$. Additionally, we present mean metrics with a 95% confidence interval (CI) and use Cohen's d to determine the effect sizes of significant differences. Comprehensive statistical analysis results can be found in the supplementary materials.

5.1 Quantitative Results

All participants successfully completed the study tasks, resulting in no variance in accuracy metrics. In the rest of this section, we present results from other measurements that highlight performance differences among the conditions.

Completion time for the navigation task. The computing environment significantly influenced the time taken for deletion (one-stop navigation) and relocation (two-stop navigation), with both $**$. On average, VR only required nearly half the time compared to DESKTOP, exhibiting statistical significance and large effect sizes. To be more specific, DESKTOP witnessed a considerable 38.0% increase in completion time (from avg. 34.7s to avg. 47.8s) between deletion and relocation, VR exhibited only a 15.7% increase (from avg. 18.8s to avg. 21.7s) (refers to Fig. 5). Consequently, we accept H_{nav} .

Completion time for the comparison task. We found that having the branching feature had a significant effect on time and execution number $**$, and the interaction of the computing environment and having the branching feature also had a significant effect on time and execution number $**$. We did not find the computing environment had a significant effect.

BRANCH conditions were significantly faster than LINEAR conditions, for both VR and DESKTOP $**$, with large effect sizes, see Fig. 10 (a). Additionally, having BRANCH also significantly reduced the number of executions required for both VR and DESKTOP, with large effect sizes, see Fig. 10 (b). Thus, we accept $H_{comp-branch}$. On the other hand, we did not observe VR outperforming DESKTOP in the comparison task. In fact, VR+BRANCH took longer than DESKTOP+BRANCH. Therefore, $H_{comp-env}$ cannot be accepted.

Text input time analysis. In our observations, we noted that participants devoted a considerable amount of time to text input within the VR environment, despite our efforts to streamline and enhance the text input experience, as outlined in Sec. 3. To systematically understand this influence, we analyzed the duration dedicated to text input across all test conditions. We discovered that the VR conditions necessitated significantly more time for text input compared to the DESKTOP conditions $**$, with VR+LINEAR taking more time than VR+BRANCH $**$ —both findings having large effect sizes. On average, text input consumed 144.3s, or 29.5% of the total time, for VR+LINEAR, and 79.3s, or 27.1% for VR+BRANCH. In contrast, desktop scenarios required a mere 7.2s (1.6%) for DESKTOP+LINEAR and 5.9s (2.3%) for DESKTOP+BRANCH, as shown in Fig. 10 (c). This data validates our observations, highlighting that text input in VR significantly hindered performance.

In a subsequent post hoc analysis, we excluded text input durations from all conditions and re-conducted the analysis, refer to Fig. 10 (d). The results suggested that VR+LINEAR was faster than

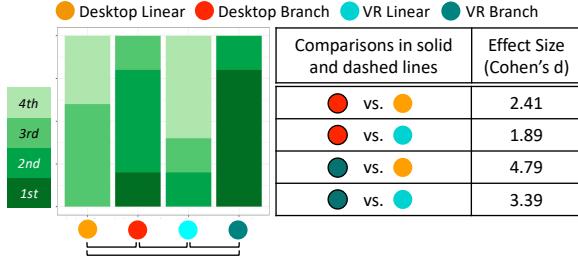


Figure 7: User ranking of overall user experience. Solid lines indicate significant differences with $p < 0.05$. The tables on the right show the Cohen's D effect sizes for significant comparisons. Circles with black borders indicate the condition with better results.

DESKTOP+LINEAR, with a medium effect size and statistical significance (**). Although there appeared to be a trend with VR+BRANCH outpacing DESKTOP+BRANCH, this distinction was not statistically significant.

Branch creation efficiency. To further investigate the performance differences between DESKTOP+BRANCH and VR+BRANCH, we examined the time taken for branch creation in both DESKTOP and VR environments, where we consider the total time of creating and placing branch windows, as illustrated in Fig. 10 (e, f). This analysis encompassed the duration required for both the creation and positioning of the newly created branches. Our findings indicate that branch creation in VR+BRANCH was notably slower than in DESKTOP+BRANCH (**), exhibiting a large effect size.

Ranking and ratings. Participants significantly favored the BRANCH conditions over the LINEAR conditions (**), as depicted in Fig. 7. Among them, VR+BRANCH was most frequently cited as offering the best overall experience, with 80% of participants placing it first. In contrast, 20% of participants ranked DESKTOP+BRANCH as their top preference, and no participants considered LINEAR conditions as their most preferred condition.

Significant effects were also observed across all perceived metrics: *physical demand* (**), *mental demand* (**), *engagement* (**), and *effectiveness* (**). BRANCH conditions outperformed LINEAR conditions in terms of mental demand (**), engagement (**), and effectiveness (**), all exhibiting large effect sizes. Moreover, VR conditions were found to be more physically demanding than DESKTOP conditions (**). VR+BRANCH emerged as the most engaging condition, boasting an average rating of 6.65 out of 7 with a confidence interval of 0.46. Impressively, 85% of participants awarded it the highest engagement score.

5.2 Layout Strategies

To better understand how participants utilized the display space, we analyzed the final layouts of each trial to understand participant strategies in Desktop and VR settings, with detailed collections available in the supplementary materials.

Within the LINEAR conditions, we observed a limited number of layout-related interactions. Notably, in DESKTOP+LINEAR, the vast majority of participants refrained from repositioning or resizing windows. Meanwhile, in VR+LINEAR, certain alterations to the initial layout were made: eight participants expanded the last result window, while two relocated select “key” windows—namely, those

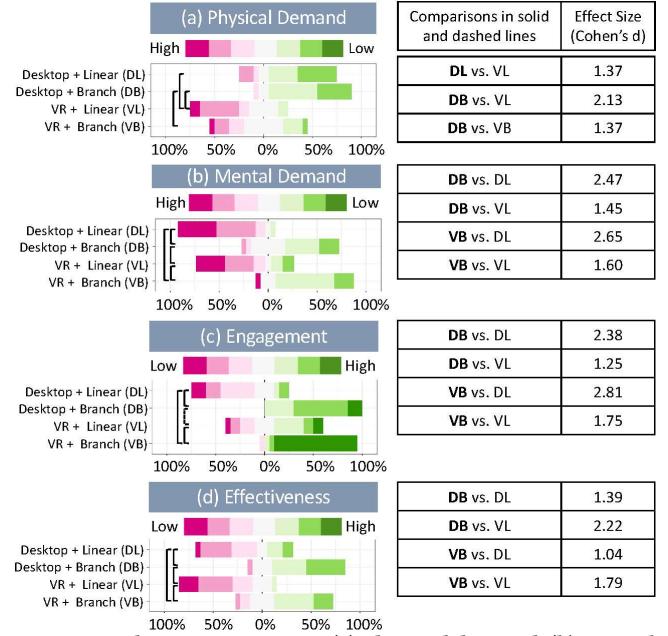


Figure 8: Subjective ratings on (a) physical demand, (b) mental demand, (c) engagement, and (d) effectiveness by task. Towards the right end of subfigures means better-perceived results. Solid lines indicate statistical significance with $p < 0.05$, and dashed lines indicate $0.05 < p < 0.1$.

containing cells crucial for generating comparisons, and the result window itself.

In contrast, the BRANCH conditions manifested a wider array of layout strategies, especially in terms of the placement of newly created branching windows. Within the DESKTOP+BRANCH condition, a dominant group of participants (15) arranged branching windows in a grid pattern. Initially, many aimed for a horizontal alignment, but due to spatial constraints, opted for additional rows, as depicted in Fig. 9 (a). The other five participants chose a vertical layout, establishing a secondary column, as illustrated in Fig. 9 (b). In VR+BRANCH, similar to VR+LINEAR, numerous participants enlarged and/or relocated particular “key” windows. Within the large display space in VR, positioning of all branching windows could be orthogonal to the initial setup direction, eliminating overlap between connecting lines and windows—a choice made by 13 participants, showcased in Fig. 9 (c). Meanwhile, seven participants opted to conserve vertical space, forming an extra column within a grid layout, displayed in Fig. 9 (d).

5.3 Qualitative Feedback

We conducted a qualitative analysis to identify common themes in user feedback for each condition. To systematically interpret our collected feedback data, two authors formulated a coding scheme rooted in the first five participants’ feedback, which was subsequently applied uniformly to subsequent participants. The top three codes and those mentioned by participants more than five times were reported for every condition (with frequency shown in parentheses). We concluded by summarizing the overarching insights

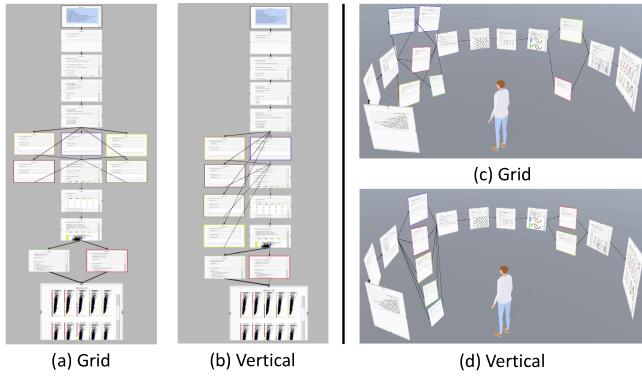


Figure 9: Two layout strategies used by our participants in DESKTOP+BRANCH(Left) and in VR+BRANCH(Right).

gleaned from all the conditions. Comprehensive coding results are available in the supplemental materials.

☒ **Desktop + I Linear** described favorable opinions with distribute cells into *multiple windows are better than a single document* (7) and *familiar* to typical computational notebook applications (4). However, drawbacks were noted, with concerns such as the need to *scroll a lot* (14), *text and target objects are hard to find* (8). These issues lead to *inhibit task performance* (5).

☒ **Desktop + I Branch** was considered positively, with *easy to compare* the results of different parameter values (18), *scroll less* (6), and *fast* to perform comparisons (6). With the integration of the branch and merge feature, participants found it to *lower the mental demand* and serve as an *effective fine-tuning* method, according to three users. However, some concerns were also noted, including that *the initial linear layout is not effective* (9), and *target objects were hard to find* (6).

☒ **VR + I Linear** was praised for its inclusion of gesture interaction and physical navigation, with descriptions such as *intuitive* (12), *easy navigation* (9), and more *fast* and *effective* than WIMP (7). Participants also found it *easy to understand the code* (7). On the other hand, major issues were related to the *text input difficulty* (14), with users stating that the provided virtual keyboard was challenging to interact with.

☒ **VR + I Branch**, similar to VR+LINEAR, was considered as *intuitive* (11), characterized by an *effective initial layout* (11), and *easy to understand the code* (4). It also demonstrated that the system provided a *easy to compare* (9), *easy navigation* (9), and was *fast* to identify the optimal result (8). Unlike DESKTOP+BRANCH, participants in this condition described it as *easy to organize* (8), with a *large display place* (7), and found managing windows to be very *flexible* (4). Notably, no specific navigation issues were identified while performing tasks in VR+BRANCH.

6 FINDINGS AND DISCUSSIONS

Our primary objective is to validate the advantages of VR over the conventional desktop computing environment. While there's a growing interest in integrating VR/AR into data analytics, as highlighted in a recent state-of-the-art report [19], empirical studies directly contrasting VR with traditional environments remain limited. Accordingly, our research seeks to offer empirical insights into both the performance and user experience distinctions between

VR and DESKTOP in the context of computational notebooks. We also aim to present quantitative data regarding the efficacy of our refined “branch & merge” design. Our subsequent findings and discussions will center on these two focal areas.

6.1 Is VR beneficial?

Our study results reveal that VR markedly surpassed DESKTOP in the navigation task. However, this advantage did not extend to the comparison task, primarily due to VR's inefficient text input mechanism.

In terms of *navigation*, participants operating within VR completed tasks faster than those using DESKTOP, both for deletion (one-stop navigation) and relocation (two-stop navigation). During one-stop navigation, participants scanned and traversed the notebook windows to locate a target. In the VR environment, this often involved head rotation, whereas in the DESKTOP environment, mouse scrolling was performed. Data suggests that physically rotating the head is more efficient than employing a mouse for navigating expansive information spaces that exceed the display size. This finding aligns with a study by Ball et al. [5], which demonstrated the efficiency of physical navigation.

We also intentionally tested a two-stop navigation, where we still found VR to be faster compared to DESKTOP. Participants, during this portion of the task, first located a target and subsequently repositioned it. We anticipated that the consistent spatial environment in VR enhances participants' spatial recall, allowing them to identify the second target more efficiently than in DESKTOP. This would lead to a relatively smaller increase in completion time and ratio. Our reasoning aligns with prior research that examined the efficacy of spatial memory within VR [36, 69].

In summary, for *navigation*, we found that VR outperformed DESKTOP within computational notebooks, largely attributed to its accelerated browsing speed and enhanced spatial awareness and memory.

In terms of *comparison*, however, the pattern shifted. While DESKTOP+LINEAR and VR+LINEAR exhibited comparable completion times, VR+BRANCH lagged behind DESKTOP+BRANCH. As delineated in Sec. 5.1, we foresaw—and subsequently confirmed—that challenges associated with text input considerably impacted the performance under VR conditions. After accounting for text input durations, VR+LINEAR significantly outperformed DESKTOP+LINEAR, utilizing its navigational strengths. In the LINEAR conditions, the effectiveness of comparisons was closely tied to navigation efficiency since participants could view only one result at a time. However, the physical navigation capabilities of VR had a positive effect on comparison tasks in these conditions. For the BRANCH conditions, the use of an intuitive VR gesture for creating branches was designed to make the process easier. Additionally, the larger display area offered by VR allowed for the simultaneous viewing of all results, which expedited the process of visual evaluation. Yet, VR+BRANCH only slightly edged out DESKTOP+BRANCH, not fully capitalizing on its navigational strengths.

Delving deeper into underlying factors, we concentrated on elements of the task not inherently tied to navigation. A detailed analysis of our interaction data unexpectedly revealed that participants took, on average, twice as long to generate a branch in

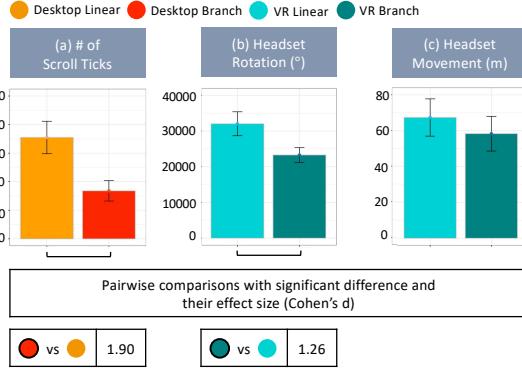


Figure 10: Navigation distance of different navigation methods across DESKTOP and VR: (a) the number of scroll ticks on the DESKTOP, (b) the degree of head rotation, and (c) the distance traversed in VR. Solid lines indicate statistical significance with $p < 0.05$. The tables below show the effect sizes for pairwise comparison. Circles with black borders indicate less navigation distance required.

VR compared to DESKTOP, see Fig. 10(e and f). We hypothesize that this increased duration is attributable to the extended physical movements required by VR’s gestural interface, a notion supported by Fitts’s law given the greater overall movement distance in the VR environment. Additionally, we observed specific behavioral patterns among participants using VR. Typically, they would first grasp the window, retreat a step to initiate the branching process, and then advance to position the resultant windows. This step-back-and-forward motion appears to be a deliberate strategy to prevent the new branch windows from colliding with existing notebook windows, while also maintaining a consistent depth of window placement in space. These additional interactions and subsequent fine-tuning of window positions incurred further time penalties in VR. Such findings are consistent with previous studies that have reported similar observations [4, 26, 66]. Nevertheless, it’s worth noting that despite the additional time required for the comparison task, participants expressed a clear preference for VR’s embodied interaction design, which received the highest ratings for both overall user experience and engagement.

In summary, for comparison, VR maintains its navigational advantage, as navigation remains a crucial element in this task. However, the system’s inefficiency in text input negatively impacted its overall performance. Additionally, although the embodied gesture interface in VR enhances user experience, it comes at the cost of increased task completion time.

6.2 Is “branch & merge” beneficial?

Yes, in our analysis, we found that the introduction of the BRANCH feature considerably shortened the completion time for comparison tasks in both VR and DESKTOP. Additionally, participants evidently perceived the BRANCH feature positively in terms of *mental demand, engagement, and effectiveness*, as well as the *overall user experience ranking*. This subjective feedback aligns closely with findings from Weinman et al. [68], wherein participants rated a similar desktop “branch” implementation. Our contributions extend this understanding by providing quantitative measures: for the comparison task

in our study, the “branch & merge” feature nearly reduced half of the completion time compared to its absence. Although creating a branch may initially take extra time, incorporating features like merging after branching can significantly reduce visual clutter and organize results spatially. This approach greatly decreases the amount of navigation effort needed. Further post hoc analysis of the navigation methods in DESKTOP and VR validated our observations: both mouse scrolling and head rotation distances were notably shorter in BRANCH than in LINEAR as shown in Fig. 10. To conclude, the “branch & merge” feature enhances the comparison process, and our study presents no evident drawbacks related to its use.

7 GENERALIZATIONS, LIMITATIONS, AND FUTURE WORK

Generalizations. Regarding *navigation*, we consider that the benefits we identified from physical navigation in VR may extend to a wide range of applications in immersive environments. This is attributable to the native support for head rotation and physical walking by the spatial tracking capabilities of VR/AR platforms. Our findings appear particularly relevant to VR/AR applications where users interact with multiple spatially-arranged windows, such as documents [14], images [44], data tables [26], maps [62], and a mixed of applications [3]. For more egocentric experiences, where a user is in a singular, immersive scene [35, 70, 73], our insights on navigation could still retain some relevance. However, the potential for increased occlusion in these views calls for further research.

Concerning *comparison*, the “branch & merge” method proved effective in both Desktop and VR environments, with the potential for beneficial integration into data flow frameworks [75, 76], no-code platforms [13, 33], and interactive visual programming [16]. These systems typically employ a graph metaphor, wherein nodes denote data or functions, and links bridge the output of one node to the input of another. This metaphor, mirroring the interconnected windows in our computational notebooks, naturally supports the essential “branch & merge” principles of code reuse maximization and simplifying logic. As parameter spaces evolve in complexity, the “branch & merge” functionality holds significant promise in facilitating hypothesis testing.

Text interaction in VR. Our study underscored challenges associated with text interactions in VR, encompassing issues like text selection, defining the entry point, and the actual typing process. On an optimistic note, the significance of enhancing text interaction for VR productivity tasks has gained consensus, and as such, it’s an evolving research domain [17, 32]. Several innovative solutions tailored for stationary environments have been presented, such as tracking physical keyboards [46] or emulating keyboards on flat surfaces. However, these challenges amplify when one introduces movement within the VR space. Advancements in sensory technologies and hardware, including haptic gloves and enhanced finger tracking, are likely to refine the VR text interaction experience in dynamic settings in the foreseeable future. An interim solution might involve minimizing mandatory text interactions. Employing

input widgets, such as dropdown menus and sliders—features commonly found in data flow systems [7, 48] or no-code data science tools [50, 58]—could serve this purpose.

Embodied gesture in VR. Our study revealed that while embodied gestures in VR enhanced the user experience, they also necessitated a longer execution time. Several factors might account for this extended duration: the greater movement distance, the need for precise placement adjustments, and efforts to prevent interference with other visual elements. The latter two challenges present opportunities for improvement. To address issues related to hand tremors or shaky mid-air gestures, future developments could incorporate a proximity snapping technique that automatically aligns the window to a predefined position as it approaches a designated area [63]. Additionally, participants in our study appeared to avoid element collisions unconsciously; future gesture design should consider this behavior. For instance, a branching gesture could be executed orthogonally to the document layout in the depth direction, thereby minimizing the risk of collision with adjacent windows. Subsequent research should validate these observations and contribute to the development of systematic guidelines for gesture design in VR.

Scalability. In a recent analysis of publicly accessible Jupyter notebooks (N=470), the study found that the average notebook comprised 125 lines of code and 20 cells [56]. The notebooks evaluated in our study were of comparable lengths. When considering the accommodation of longer notebooks, several potential strategies emerge. One approach is to extend the curvature of the layout, positioning notebooks at a greater distance from the user. However, this requires increased user movement and could compromise content readability. Alternatively, vertical space could be utilized to arrange windows in a grid format; however, this introduces the challenge of accessing elevated windows and may require additional interaction designs. In summary, future research should explore these trade-offs and consider other potential solutions for effectively accommodating longer computational notebooks and more complex branching scenarios.

Addressing Additional Computational Notebook Challenges in VR. Our study primarily aims to leverage VR for enhancing navigation and comparison in computational notebooks, as these are fundamental interactions in data analysis where VR can potentially offer significant improvements. We acknowledge that our design may not represent the optimal adaptation of the computational notebook framework in VR. Our current adaptation focuses on examining the impacts of specific factors we aimed to explore, but other innovative approaches could exist. Moreover, computational notebooks face various other challenges, like the ones identified in a previous comprehensive study: setup, exploration and analysis, managing code, reliability, archival, security, sharing and collaboration, reproducing and reusing, and notebooks as products [9]. We believe addressing those challenges will be a long-term effort from multiple communities. For example, Wang et al. [67] highlighted the possibilities for real-time collaboration among multiple users. Moving forward, we want to explore how to better exploit the unique display and interaction capabilities of VR to improve those experiences.

8 CONCLUSION

We adapted the computational notebook interface from desktop to VR and tested its effectiveness through a controlled study. Our results revealed that notebooks in VR outperformed notebooks on DESKTOP in navigation efficiency, and the inclusion of a “branch & merge” feature notably enhanced the non-linear comparison process. Participants reported that the integration of VR with the “branch&merge” functionality was the most engaging and provided the best overall user experience among all test conditions. However, we observed that text interaction in VR remains a challenge. This issue could be alleviated with future advancements in hardware and tracking technologies, and as users become more familiar with VR environments over time. Our study underscores the immense potential of computational notebooks in VR, particularly in enhancing navigation and comparison performance and experience for analysts. It’s important to note that our VR adaptation was specifically tailored to investigate navigation and comparison, and there may be other innovative approaches for adapting or completely redesigning computational notebooks in VR. Broadly, our results provide preliminary evidence supporting the wider use of large display spaces, augmented spatial awareness, embodied interaction, and physical navigation in VR for immersive analytics applications.

ACKNOWLEDGMENTS

This work was supported in part by NSF I/UCRC CNS-1822080 via the NSF Center for Space, High-performance, and Resilient Computing (SHREC).

REFERENCES

- [1] Sara Alspaugh, Nava Zokaei, Andrea Liu, Cindy Jin, and Marti A Hearst. 2018. Futzling and moseying: Interviews with professional data analysts on exploration practices. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 22–31.
- [2] Christopher Andrews, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 55–64.
- [3] Apple. 2023. *Apple VR/AR Device*. <https://www.apple.com/apple-vision-pro/> Accessed: 2023-1-15.
- [4] Benjamin Bach, Ronell Sicat, Johanna Beyer, Maxime Cordeil, and Hanspeter Pfister. 2018. The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 457–467. <https://doi.org/10.1109/TVCG.2017.2745941>
- [5] Robert Ball, Chris North, and Doug A. Bowman. 2007. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*. ACM Press, San Jose, California, USA, 191–200. <https://doi.org/10.1145/1240624.1240656>
- [6] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., USA.
- [7] Apache Camel. 2023. *Apache Camel Data Flow system*. <https://camel.apache.org/> Accessed: 2023-1-15.
- [8] CarbideAlpha. 2023. *Computational notebook tool*. <https://alpha.trycarbide.com/> Accessed: 2023-1-15.
- [9] Souti Chattopadhyay, Ishita Prasad, Austin Z Henley, Anita Sarma, and Titus Barik. 2020. What’s wrong with computational notebooks? Pain points, needs, and design opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [10] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. 2009. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)* 41, 1 (2009), 1–31.
- [11] Maxime Cordeil, Tim Dwyer, Karsten Klein, Bireswar Laha, Kim Marriott, and Bruce H Thomas. 2016. Immersive collaborative analysis of network connectivity: CAVE-style or head-mounted display? *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2016), 441–450.

[12] DataBricks. 2023. *Computational notebook tool*. <https://www.databricks.com/> Accessed: 2023-1-15.

[13] DataRobot. 2023. *No-code Data Science Platforms*. <https://www.datarobot.com/> Accessed: 2023-1-15.

[14] Kylie Davidson, Lee Lisle, Kirsten Whitley, Doug A Bowman, and Chris North. 2022. Exploring the Evolution of Sensemaking Strategies in Immersive Space to Think. *IEEE Transactions on Visualization and Computer Graphics* (2022).

[15] Robert DeLine, Andrew Bragdon, Kael Rowan, Jens Jacobsen, and Steven P Reiss. 2012. Debugger canvas: industrial experience with the code bubbles paradigm. In *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 1064–1073.

[16] Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu Liu, et al. 2023. Rapsai: Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23.

[17] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text entry in virtual reality: A comprehensive review of the literature. In *Human-Computer Interaction. Recognition and Interaction Technologies: Thematic Area, HCI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part II 21*. Springer, 419–437.

[18] Niklas Elmqvist, Nathalie Henry, Yann Riche, and Jean-Daniel Fekete. 2008. Melange: space folding for multi-focus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1333–1342.

[19] Barrett Ens, Benjamin Bach, Maxime Cordeil, Ulrich Engelke, Marcos Serrano, Wesley Willett, Arnaud Prouzeau, Christoph Anthes, Wolfgang Büschel, Cody Dunne, et al. 2021. Grand challenges in immersive analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[20] GoogleColab. 2023. *Computational Notebook Tool*. <https://colab.google/> Accessed: 2023-1-15.

[21] Jesse Harden, Elizabeth Christman, Nurit Kirshenbaum, John Wenskovitch, Jason Leigh, and Chris North. 2022. Exploring Organization of Computational Notebook Cells in 2D Space. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–6.

[22] Björn Hartmann, Loren Yu, Abel Allison, Yeonsoo Yang, and Scott R Klemmer. 2008. Design as exploration: creating interface alternatives through parallel authoring and runtime tuning. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 91–100.

[23] Devamardip Hayatpur, Haijun Xia, and Daniel Wigdor. 2020. Datahop: Spatial data exploration in virtual reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 818–828.

[24] Andrew Head, Fred Hohman, Titus Barik, Steven M Drucker, and Robert DeLine. 2019. Managing messes in computational notebooks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[25] Helen H Huang, Hanspeter Pfister, and Yalong Yang. 2023. Is embodied interaction beneficial? A study on navigating network visualizations. *Information Visualization* 22, 3 (2023), 169–185.

[26] Sungwon In, Tica Lin, Chris North, Hanspeter Pfister, and Yalong Yang. 2023. This is the Table I Want! Interactive Data Transformation on Desktop and in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[27] JupyterNotebook. 2020. *Jupyter Survey*. <https://github.com/jupyter/surveys/blob/master/surveys/2020-12-jupyter-survey/README.md> Accessed: 2023-8-31.

[28] JupyterNotebook. 2023. *Computational notebook tool*. <https://jupyter.org/> Accessed: 2023-1-15.

[29] DaYe Kang, Tony Ho, Nicolai Marquardt, Bilge Mutlu, and Andrea Bianchi. 2021. Toonote: Improving communication in computational notebooks using interactive data comics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[30] Mary Beth Kery, Amber Horvath, and Brad A Myers. 2017. Variolite: Supporting Exploratory Programming by Data Scientists... In *CHI*, Vol. 10. 3025453–3025626.

[31] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E John, and Brad A Myers. 2018. The story in the notebook: Exploratory data science using a literate programming tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[32] Pascal Knierim, Thomas Kosch, Johannes Groschopp, and Albrecht Schmidt. 2020. Opportunities and challenges of text input in portable virtual reality. In *Extended abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–8.

[33] KNIME. 2023. *Computational Notebook Tool*. <http://www.knime.org/> Accessed: 2023-1-15.

[34] Donald Ervin Knuth. 1984. Literate programming. *The computer journal* 27, 2 (1984), 97–111.

[35] M. Kraus, N. Weiler, D. Oelke, J. Kehrer, D. A. Keim, and J. Fuchs. 2019. The Impact of Immersion on Cluster Identification Tasks. *IEEE Transactions on Visualization and Computer Graphics* (2019), 1–1. <https://doi.org/10.1109/TVCG.2019.2934395>

[36] Eric Krokos, Catherine Plaisant, and Amitabh Varshney. 2019. Virtual memory palaces: immersion aids recall. *Virtual reality* 23 (2019), 1–15.

[37] Oh-Hyun Kwon, Chris Muelder, Kyungwon Lee, and Kwan-Liu Ma. 2016. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 7 (July 2016), 1802–1815. <https://doi.org/10.1109/TVCG.2016.2520921>

[38] Tica Lin, Yalong Yang, Johanna Beyer, and Hanspeter Pfister. 2021. Labeling out-of-view objects in immersive analytics to support situated visual searching. *IEEE Transactions on Visualization and Computer Graphics* (2021).

[39] Lee Lisle, Xiaoyu Chen, J.K. Edward Gitre, Chris North, and Doug A. Bowman. 2020. Evaluating the Benefits of the Immersive Space to Think. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, Atlanta, GA, USA, 331–337. <https://doi.org/10.1109/VRW50115.20200073>

[40] Lee Lisle, Kylie Davidson, Edward J.K. Gitre, Chris North, and Doug A. Bowman. 2021. Sensemaking Strategies with Immersive Space to Think. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, Lisboa, Portugal, 529–537. <https://doi.org/10.1109/VR50410.2021.00077>

[41] Jiali Liu, Nadia Boukhelifa, and James R Eagan. 2019. Understanding the role of alternatives in data analysis practices. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 66–76.

[42] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. 2020. Design and evaluation of interactive small multiples data visualisation in immersive spaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 588–597.

[43] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. 2022. Effects of display layout on spatial memory for immersive environments. *Proceedings of the ACM on Human-Computer Interaction* 6, ISS (2022), 468–488.

[44] Weizhou Luo, Anke Lehmann, Hjalmar Widengren, and Raimund Dachselt. 2022. Where should we put it? layout and placement strategies of documents in augmented reality for collaborative sensemaking. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[45] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H Thomas. 2018. *Immersive analytics*. Vol. 11190. Springer.

[46] Meta. 2023. *Meta Physical Trackable Keyboard*. <https://www.meta.com/help/quest/articles/headsets-and-accessories/meta-quest-accessories/tracked-keyboards-meta-quest/> Accessed: 2023-1-15.

[47] Arnab Nandi, Liling Jiang, and Michael Mandel. 2013. Gestural query specification. *Proceedings of the VLDB Endowment* 7, 4 (2013), 289–300.

[48] Apache NiFi. 2023. *Apache NiFi Data Flow system*. <https://nifi.apache.org/> Accessed: 2023-1-15.

[49] Niels Christian Nilsson, Stefania Serafin, Frank Steinicke, and Rolf Nordahl. 2018. Natural Walking in Virtual Reality: A Review. *Computers in Entertainment* 16, 2 (April 2018), 1–22. <https://doi.org/10.1145/3180658>

[50] ORANGE. 2023. *Computational Notebook Tool*. <https://orangedatamining.com/> Accessed: 2023-1-15.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[52] Jeffrey S Pierce, Brian C Stearns, and Randy Pausch. 1999. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*. 141–145.

[53] Matthew Plumlee and Colin Ware. 2002. Zooming, multiple windows, and visual working memory. In *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '02*. ACM Press, Trento, Italy, 59. <https://doi.org/10.1145/1556262.1556270>

[54] Matthew D. Plumlee and Colin Ware. 2006. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM Transactions on Computer-Human Interaction (TOCHI)* 13, 2 (June 2006), 179–209. <https://doi.org/10.1145/1165734.1165736>

[55] Deepthi Raghunandan, Aayushi Roy, Shenzhi Shi, Niklas Elmqvist, and Leilani Battle. 2023. Code evolution: Understanding how people change data science notebooks over time. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–12.

[56] Dhivyabharathi Ramasamy, Cristina Sarasua, Alberto Bacchelli, and Abraham Bernstein. 2023. Workflow analysis of data science code in public GitHub repositories. *Empirical Software Engineering* 28, 1 (2023), 7.

[57] Rama Rao and Stuart K Card. 1994. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 318–322.

[58] RapidMiner. 2023. *No-code Data Science tools*. <https://rapidminer.com/> Accessed: 2023-1-15.

[59] Gunnar Reiske, Sungwon In, and Yalong Yang. 2023. Multi-Focus Querying of the Human Genome Information on Desktop and in Virtual Reality: an Evaluation. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 1123–1131.

[60] Adam Rule, Aurélien Tabard, and James D Hollan. 2018. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[61] Jeffrey M Rzeszotarski and Aniket Kittur. 2014. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 897–906.

[62] Kadek Ananta Satriadi, Barrett Ens, Maxime Cordeil, Tobias Czauderna, and Bernhard Jenny. 2020. Maps around me: 3D multiview layouts in immersive spaces. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (2020), 1–20.

[63] Abhishek Seth, Judy M Vance, and James H Oliver. 2011. Virtual reality for assembly methods prototyping: a review. *Virtual reality* 15 (2011), 5–20.

[64] Aurélien Tabard, Wendy E Mackay, and Evelyn Eastmond. 2008. From individual to collaborative: the evolution of prism, a hybrid laboratory notebook. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. 569–578.

[65] Wai Tong, Meng Xia, Kam Kwai Wong, Doug A Bowman, Ting-Chuen Pong, Huamin Qu, and Yalong Yang. 2023. Towards an Understanding of Distributed Asymmetric Collaborative Visualization on Problem-solving. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 387–397.

[66] Jorge A Wagner Filho, Marina F Rey, Carla MDS Freitas, and Luciana Nedel. 2018. Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 483–490.

[67] April Yi Wang, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the "Why" by Connecting Conversations with Computational Narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[68] Nathaniel Weinman, Steven M Drucker, Titus Barik, and Robert DeLine. 2021. Fork It: Supporting stateful alternatives in computational notebooks. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[69] Fumeng Yang, Jing Qian, Johannes Novotny, David Badre, Cullen D. Jackson, and David H. Laidlaw. 2021. A Virtual Reality Memory Palace Variant Aids Knowledge Retrieval from Scholarly Articles. *IEEE Transactions on Visualization and Computer Graphics* 27, 12 (Dec. 2021), 4359–4373. <https://doi.org/10.1109/TVCG.2020.3009003>

[70] Yalong Yang, Maxime Cordeil, Johanna Beyer, Tim Dwyer, Kim Marriott, and Hanspeter Pfister. 2020. Embodied navigation in immersive abstract data visualization: Is overview+ detail or zooming better for 3D scatterplots? *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1214–1224.

[71] Yalong Yang, Tim Dwyer, Bernhard Jenny, Kim Marriott, Maxime Cordeil, and Haohui Chen. 2019. Origin-Destination Flow Maps in Immersive Environments. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 693–703. <https://doi.org/10.1109/tvcg.2018.2865192>

[72] Yalong Yang, Tim Dwyer, Kim Marriott, Bernhard Jenny, and Sarah Goodwin. 2020. Tilt map: Interactive transitions between Choropleth map, Prism map and Bar chart in immersive environments. *IEEE Transactions on Visualization and Computer Graphics* 27, 12 (2020), 4507–4519.

[73] Yalong Yang, Bernhard Jenny, Tim Dwyer, Kim Marriott, Haohui Chen, and Maxime Cordeil. 2018. Maps and Globes in Virtual Reality. *Computer Graphics Forum* 37, 3 (June 2018), 427–438. <https://doi.org/10.1111/cgf.13431>

[74] Yalong Yang, Wenya Xia, Fritz Lekschas, Carolina Nobre, Robert Krüger, and Hanspeter Pfister. 2022. The pattern is in the details: An evaluation of interaction techniques for locating, searching, and contextualizing details in multivariate matrix visualizations. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.

[75] Bowen Yu and Cláudio T Silva. 2016. VisFlow-Web-based visualization framework for tabular data with a subset flow model. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2016), 251–260.

[76] Bowen Yu and Cláudio T Silva. 2019. FlowSense: A natural language interface for visual data exploration within a dataflow system. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 1–11.

[77] Zeppelin. 2023. *Computational notebook tool*. <https://zeppelin.apache.org/> Accessed: 2023-1-15.

[78] Yuheng Zhao, Jinjing Jiang, Yi Chen, Richen Liu, Yalong Yang, Xiangyang Xue, and Siming Chen. 2022. Metaverse: Perspectives from graphics, interactions and visualization. *Visual Informatics* 6, 1 (2022), 56–67.