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

Kirsten Whitley
Department of Defense
United States

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) × 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.

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.

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. Cokic 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]. 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
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...

Figure 3: Illustrations of gestures interactions in the VR environment.
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 fingers (vs. mouse to select the target and click the buttons to execute specific commands, like extract, delete, and relocate. To ensure an equitable comparison and to control for potential confounders, we integrated features detailed in Sec. 3.

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 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 commands, like extract, delete, and relocate.

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 [31]. 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 x 1920. For Desktop conditions, a standard 27” monitor was deployed, featuring a 2560x1440 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 16m² 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 0.35x0.30m², 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 x 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\textsubscript{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\textsubscript{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.

### 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 $p < 0.001$ (***)
Table 1: Comparisons of performance metrics across conditions. 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。

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 $H_{\text{nav}}$.

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 $H_{\text{comp} - \text{branch}}$. 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, $H_{\text{comp} - \text{env}}$ 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
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 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, option 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 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.

**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 $0.05 < p < 0.1$. 

<table>
<thead>
<tr>
<th>Comparisons in solid and dashed lines</th>
<th>Effect Size [Cohen’s d]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop + Linear (DL) vs. Desktop + Branch (DB)</td>
<td>2.41</td>
</tr>
<tr>
<td>Desktop + Linear (DL) vs. VR + Linear (VL)</td>
<td>1.89</td>
</tr>
<tr>
<td>VR + Linear (VL) vs. VR + Branch (VB)</td>
<td>4.79</td>
</tr>
<tr>
<td>VR + Branch (VB) vs. VR + Linear (VL)</td>
<td>3.39</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparisons in solid and dashed lines</th>
<th>Effect Size [Cohen’s d]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB vs. DL</td>
<td>1.04</td>
</tr>
<tr>
<td>DB vs. VL</td>
<td>1.25</td>
</tr>
<tr>
<td>VB vs. DL</td>
<td>2.81</td>
</tr>
<tr>
<td>VB vs. VL</td>
<td>1.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparisons in solid and dashed lines</th>
<th>Effect Size [Cohen’s d]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL vs. VL</td>
<td>1.37</td>
</tr>
<tr>
<td>DB vs. VL</td>
<td>1.37</td>
</tr>
<tr>
<td>VB vs. DL</td>
<td>1.79</td>
</tr>
</tbody>
</table>
gleaned from all the conditions. Comprehensive coding results are available in the supplemental materials.

**Desktop + 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 + 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 + 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 + 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+Branch 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 VR will still find 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
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 the VR interface in 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 these 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**


