Exploring Organization of Computational Notebook Cells in 2D Space

1st Jesse Harden
Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
jessemh@vt.edu

2nd Elizabeth Christman
Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
echristman99@vt.edu

3rd Nurit Kirshenbaum
Dept. of Information and Computer Sciences
University of Hawaii at Manoa
Honolulu, HI, USA
nuritk@hawaii.edu

4th John Wenskovitch
Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
jw87@vt.edu

5th Jason Leigh
Dept. of Information and Computer Sciences
University of Hawaii at Manoa
Honolulu, HI, USA
leighj@hawaii.edu

6th Chris North
Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
north@vt.edu

***Abstract*—Representing branching and comparative analyses in computational notebooks is complicated by the 1-dimensional (1D), top-down list arrangement of cells. Given the ubiquity of these and other non-linear features, their importance to analysis and narrative, and the struggles current 1D computational notebooks have, enabling organization of computational notebook cells in 2 dimensions (2D) may prove valuable. We investigated whether and how users would organize cells in such a "2D Computational Notebook" through a user study and gathered feedback from participants through a follow-up survey and optional interviews. Through the user study, we found 3 main design patterns for arranging notebook cells in 2D: Linear, Multi-Column, and Workboard. Through the survey and interviews, we found that users see potential value in 2D Computational Notebooks for branching and comparative analyses, but the expansion from 1D to 2D may necessitate additional navigational and organizational aids.**

***Index Terms*—programming, data science, software, computational notebooks**

I. Introduction

Computational notebooks are popular tools for data analysis and presentation [1]. Since open-sourced notebooks such as Jupyter [2] emerged, millions of users from diverse fields have adopted computational notebooks for their work [3]; entire research communities, such as the astronomy community, adopted computational notebooks for their work [4]. Computational notebooks combine code, visualizations, and text into a single document, which enables analysts to construct and present a *computational narrative* [3], a reproducible, collaboratively created narrative document that tells a story for a particular set of audiences and contexts [5]. Computational notebooks also allow interleaving of results with code and in-place editing of code, which empowers analysts during the iterative exploratory process of data analysis [3], [6] to quickly test and refine models on their data and see the results.

This research was supported in part by NSF grant #CSSI-2003387.

Current computational notebooks are not without limitations; one such limitation is how they handle non-linearity in both analysis and narrative presentation given the 1-dimensional (1D) layout of cells. Rule, Tabard, and Hollan [3] view support for non-linear narratives as a design opportunity for computational notebooks. We identify two common types of non-linearity: different operations on same data, and same operations on different data. The former may occur when an analyst tries different models, tweaks parameters, and compares results to iteratively refine their work on a single dataset. This may be valuable for presentation to a technical audience interested in why a certain algorithm was used instead of others. It can also occur when plotting different graphs of the data for exploration or analysis. The latter may occur when an analyst performs the same analytic steps on two different sets of data, such as separate subsets of a single, larger dataset; this may be valuable for comparing results.

Another limitation of current computational notebooks is navigating long notebooks which contain many cells. The vertical, 1D layout of a notebook, traversed through scrolling up and down, is tedious to navigate with a long notebook; finding a code fragment, a common task for debugging, is difficult without memory of its location. To address this issue, some authors take advantage of large screen space by opening the same computational notebook in multiple side-by-side windows, with each one showing a different part of the notebook. While this method is helpful, its necessity suggests a need for improvement.

To better represent non-linearity and improve navigation in longer notebooks, the authors propose organizing computational notebook cells in 2 dimensions (2D); instead of ordering cells from top to bottom as an ordered list, a 2D environment may empower users to address non-linear narratives with non-linear configurations and enable additional methods of navigation in addition to scrolling up and down. The additional space may also empower users to encode more meaning into the space, much like how analysts used additional screen space in studies of Space to Think [7], [8].

This paper contributes to ongoing research on computational notebooks through the novel idea of a 2D Computational Notebook, an exploration of the potential use and value of 2D space for organizing computational notebook cells, as well as requirements gathering and considerations for the potential design of 2D notebooks.

To accomplish this, we focus on the following research questions:

1) Given a notebook with non-linear features, would users utilize 2D space in their notebook organization?
2) How would users organize the notebook cells in 2D space?
3) How would users encode the run order of cells into the notebook layout in 2D space?
4) What strengths and weaknesses might 2D notebooks have compared to 1D notebooks?
5) Would users be interested in using 2D computational notebooks in their work?

To answer these questions, we conducted a study where users organized images of computational notebook cells on a Miro board [9], an online infinite 2D canvas website, which was followed up by a survey and an optional interview. We learned that users utilized 2D space for computational notebooks with a few distinct approaches, each enabling intuitive understanding of the desired run order. These approaches could lead to a set of common design patterns or templates for future notebook tools. Furthermore, we found that 2D computational notebooks may benefit tasks and narratives with comparative analysis or branching paths, but that navigation and cell organization may continue to be tedious without effective aids.

II. Background and Related Works

This research builds on three bodies of related work: the design of computational notebooks, computational narratives, and the use of space in analysis.

A. Design of Computational Notebooks

The design of today's computational notebooks, such as Jupyter Notebooks [2], was influenced by Knuth's concept of *literate programming* [10], in which an author "weaves human language with live code and the results of the code" to produce a computational narrative [5]. To this end, computational notebooks address diverse challenges of data analysis. Support for incremental and iterative analysis, rich explanation of an analyst's thoughts and processes, and sharing of code, text, and visuals in a single document [3] make computational notebooks an excellent tool for data analysis.

However, computational notebooks do present some struggles, or pain points, for users; Chattopadhyay et al. [11] list several pain points for users of computational notebooks, which includes exploration and analysis.

For example, one major struggle with the iterative process of exploration and analysis is how messy notebooks can become [3], [12], [13]. Analysts have described notebooks and their code as "ad hoc" or "throw-away" [6], [14] and in need of cleaning [3] before being ready for presentation. Part of the reason for this messiness, as found by Kery et al. [13] is that data analysts, as part of the process of exploring alternatives, replicate code across many cells that must later be refactored, a process that is both tedious and error-prone [13], [15]. In short, the process of exploration and analysis appears at times to run counter to the goal of developing a clean and coherent computational narrative.

Some proposed solutions to this issue include enabling forking and backtracking of stateful alternatives [15] and version control systems for computational notebooks [16], [17]. Of note is that Weinman's work on forking [15] does introduce a constrained use of 2D space, which may support the idea of using 2D to address computational notebook issues.

Another approach to addressing such issues for computational notebooks is the development of best practices such as documenting every exploration, as Rule et al. [3], [18] encouraged and proposed. Support for following best practices can resolve many issues; we support such work to identify and encourage use of best practices, and do not think such guidelines detract from the potential of 2D space for organizing computational notebooks. Rather, we suggest that these two solutions are complementary, and appropriate use of 2D space could itself be a form of future best practice.

B. Computational Narratives

One of the strengths of computational notebooks are their ability to combine code, text, and visualizations to form computational narratives, a series of ordered and connected events related to computational analyses; this ability is useful because humans process information better as stories or narratives [3].

As noted by Perez & Granger [5], computational narratives are developed for a particular set of audiences and contexts; given different audiences and contexts, different storytelling strategies may be necessary [19]. For an audience that just wants to see the results of an analysis, a completely linear narrative with only those cells relevant to the final outputs can make sense. However, some tasks that such an audience may want to do, such as comparing results between different subsets, may benefit from a non-linear narrative structure. Weinman et al. [15] found that "the ability to proximally compare" different visual representations of data was critical to analysis processes. In addition, an audience that wants to understand and evaluate the entire analytic process may find a non-linear narrative structure better for the task, as it can expose alternatives tried and compared to the final results used, enhancing reproducibility. Weinman et al. [15] found that analysts used forking paths to compare results of machine learning models as part of their exploratory analysis; while their work investigated forking as an analysis tool, the ability to see what options an analyst tried, as well as the results of said options in a proximally comparable way, could be a powerful feature of non-linear narratives structures.

C. Utilizing Space in Analysis

Enabling users to effectively utilize space for analysis tasks is an ongoing area of research. Two relevant research threads in this area include Space to Think and Code Bubbles.

1) Space to Think: Andrews, Endert, and North [7] studied the use of large, high-resolution displays for sensemaking, and found that the additional space provided by the displays aided users in two particular ways: first, it enabled users to externalize memory, which allowed users to focus more on the task at hand rather than on recalling important info; second, it enabled users to encode meaning into the space, such as by clustering similar information together. This work has since been expanded through study of the additional space provided by virtual reality devices [8], as well as collaborative uses of large spaces through increased and varied content contribution [20].

2) Code Bubbles and VisSnippets: A similar line of research exploring the use of space in programming would be Code Bubbles by Bragdon et al. [21] and VisSnippets by Burks, Renambot, and Johnson [22].

Code Bubbles [21] is an approach for integrated development environments (IDEs) that uses the *bubble* metaphor; a bubble is an interactive and editable fragment of such items as code, documentation, or debugging displays. In addition, bubbles can be clustered together to represent a "concurrently visible working set." Their work found similar benefits for the expanded use of space in Code Bubbles as Andrews, Endert, and North [7] found for the use of large displays. Specifically, Bragdon et al. [21] found that developers appreciated being able to externalize information from their "limited working memory" through "bubbles and annotations." In addition, they found that the ability to position bubbles strategically made developers more efficient through the use of "spatial proximity and spatial memory."

VisSnippets [22] is a system designed for collaborative data exploration that uses blocks of reusable code called "snippets" connected via arrows to form a 2D visual dataflow display; this system empowers quick exploration of "complementary and contrasting analyses" through reuse of snippets a group has made over time.

Both Code Bubbles and VisSnippets design 2D spaces for coding; our work expands by considering the use of 2D space for computational narratives and studying its usage.

III. Methodology

To answer our research questions, we designed a study consisting of a screening questionnaire, a user study task, a post-task survey, and an optional interview. We designed the user study task to answer research questions 1-3; participants organized cells of a computational notebook using a layout strategy of their choosing in this part. For the post-task survey and optional interview, we focused on research questions 4 and 5.

A. Recruitment and Screening

We recruited 50 participants via academic listservs of students and faculty from two universities. Each participant completed a screening questionnaire asking whether they had experience with both Python and computational notebooks such as Jupyter. We selected 43 participants with experience in both tools to continue; 25 volunteered to complete the study. Next, we invited 6 participants to take part in a semi-structured interview; those invited used different strategies in their 2D computational notebook cell layout or provided thoughtful survey responses worth exploring further. Of those invited, 5 were interviewed.

B. User Study Task

For this study, we created a computational notebook that consists of an analysis of publicly available COVID-19 data; the notebook focused on Virginia overall and two of its counties: Fairfax and Henrico. Knowledge of Virginia was neither required nor expected from participants. The analysis contained several non-linear features; it included three different charts for Virginia overall, showing the same data with different analyses, while the two counties had the same analyses done with different data; each county had 3 graphs, 2 of which were predictive. Furthermore, the analysis was divided into sections, with markdown cells noting the beginning of each section.

Each notebook cell (code cell along with its output view) was converted into an image and randomly placed into a jumbled pile at the center of a Miro Board [9]. Miro Boards provide an infinite 2D canvas that allow users to move images around at will, connect them with arrows, create labelled frames that images can be put onto, and add sticky notes with text, among other features. We chose to use Miro Boards in part because of these additional features that are not currently present in computational notebooks to explore what kinds of additional visual features might be relevant to the design and implementation of 2D computational notebooks.

Fig. 1. User Study Task Starting Point

We gave each participant access to a personal Miro Board arranged as seen in Figure 1 and instructed them to take no more than 40 minutes to complete the user study task described below:

> The Miro Board has a collection of 26 cells from a Jupyter Notebook for analysis of COVID-19 data scrambled into a pile. Your task is to take the cells in the notebook and organize them in 2D space as if you were going to present the notebook layout to a new hire on your data analysis dev team who is looking to learn not only the results of the analysis, but also the process through which the results were developed, and will continue developing the notebook with you. You may not only move cells around, but also use whatever tools the Miro Board provides that help you make a compelling, useful layout.

We designed this prompt to set presentation and development as key considerations for participants as they created their visual layouts. Unfortunately, due to COVID-19 and our desire to enable as many participants to perform the study as possible, we were unable to view participants while they completed the task.

To analyze the visual layouts created, we accumulated all of the layouts created by participants along with their minimaps and put them into a single Miro board for comparison and pattern finding. Comparison and pattern finding was done with an open coding approach with multiple coders.

C. Post-Task Survey

We instructed participants to complete the post-task survey after they finished the user study task. This survey consisted of 19 seven-point Likert-Scale (strongly disagree to strongly agree) questions investigating participants' attitudes towards the potential of 2D Computational Notebooks, as well as several qualitative questions asking about the visual layout they created, why they created the layout they did, and their initial thoughts on the potential of 2D Computational Notebooks. We analyzed the qualitative data using open coding to identify themes, and the Likert-Scale data using frequencies for each choice; these frequencies were then used to calculate the mean and median responses. Since Likert-Scale data is technically ordinal and thus not truly quantitative, we include the median as a potentially more reliable measure.

D. Semi-Structured Interviews

For the interviews, we delved further into how each participant organized their layout, the reasoning behind their layout, and potential benefits and tradeoffs of 2D computational notebooks as compared to 1D computational notebooks. Each interview lasted 45 minutes to 1 hour. We transcribed each interview and used open coding to identify themes.

IV. Results

A. User Study Task Results

Through analyzing the 25 different visual layouts, summarized in Figure 2, we answered research questions 1-3. Most participants utilized 2D space in their layout, with the layouts grouped into 3 distinct approaches. Furthermore, the authors easily interpreted the run order of all participants' layouts except for the layout by participant P09.

1) High-Level Design Patterns: We identified 3 high-level design patterns to organizing the notebook cells in 2D: Linear (7 instances), Multi-Column (8 instances), and Workboard (10 instances). Here we describe each of these approaches and their subgroups, where applicable.

The **Linear** design pattern is defined by its use of a single column of cells as the backbone of the computational notebook layout. This approach had three subgroups: Traditional, Split-Cell, and Split-Column. Traditional Linear (4 instances) is equivalent to a 1D Computational Notebook in layout. Split-Cell Linear (2 instances), as seen in Figure 3 is nearly equivalent to a 1D Computational Notebook except that at least 1 cell in the column is "split" into a row of cells; this is similar to the Split Cells extension [23] for Jupyter Notebooks except that the row of cells can extend beyond the width of one cell. Split-Column Linear (1 instance) starts with a single column as the backbone before splitting into two or more columns, much like Fork-It [15].

The run order of Linear layouts was always top-to-bottom, with Split Cells and Split Columns being run left-to-right. Depending on the cells or columns split, however, running in any order or in parallel could be possible and may represent cognitive branching.

The **Multi-Column** design pattern arranged cells in columns that were ordered from left to right. The linear columns represented "chunks" or sections from the overall linear notebook, each chunk with its own train of thought. Unlike the Split-Column Linear approach, all columns are top aligned, starting at roughly the same y position at the top of their layout, instead of branching off from an initial column. As an exception to this rule, 5 users aligned two columns containing the county analyses, thus pushing the second county column slightly lower due to the markdown cells at the top of the first (as in Fig 4). The run order for this approach was universally column-major order: top-to-bottom linear within each column, and left-to-right flow of columns.

The **Workboard** design pattern involved more complex 2D layouts than the Linear and Multi-Column design patterns, and had two sub-groups: Grouped Combinations and Directed Graphs. The Grouped Combinations (6 instances) organize the notebook cells into sections, which may differ in strategy (e.g. Linear, Multi-Column), and the sections are also organized according to a strategy. For example, sections may be arranged in a linear, top-down fashion while each section adopts a multi-column, left-to-right approach. Participant P09, the only participant whose run order was not intuitive, was classified as part of this; they appeared to arrange sections in a top-down manner and may have meant for sections to run in a clockwise rotational manner, starting with the upper left code cell. The Directed Graph approach (4 instances), on the other hand, uses arrows to develop more complex, flowchart-like run orders (as in Fig 5). These flows are similar to node-link diagrams which

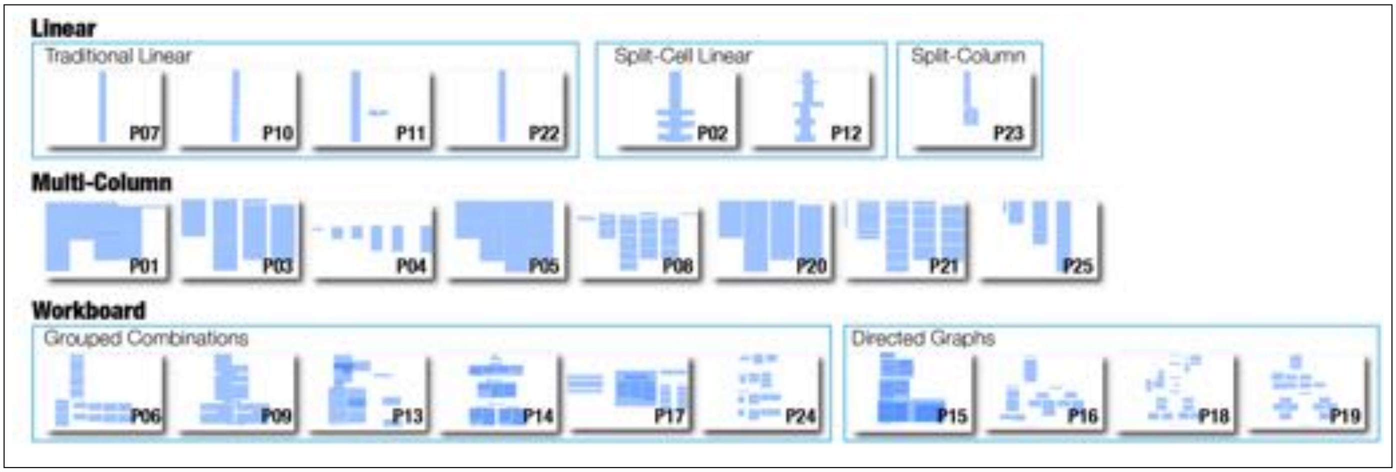

Fig. 2. Minimaps of User Study Task Results

Fig. 3. Example Linear design pattern with 3 split cells.

have found use in visual programming endeavors like the work of Suzuki et al. with block-based programming [24]. Each of the 4 Directed Graph instances used a top-down progression, with 2 also sticking to a left-to-right progression.

2) Representations of Non-Linearity: In addition to analyzing the different visual layout approaches, we also took note of whether and how participants utilized 2D space to display non-linear features, such as different data, same analysis, of the computational notebook. Of the 25 participants' layouts, 20 layouts appeared to utilize the 2D space for at least one kind of non-linearity mentioned in this paper's introduction. For

Fig. 4. Example Multi-Column design pattern, with parallel county analyses in the right-most two columns to enable comparison.

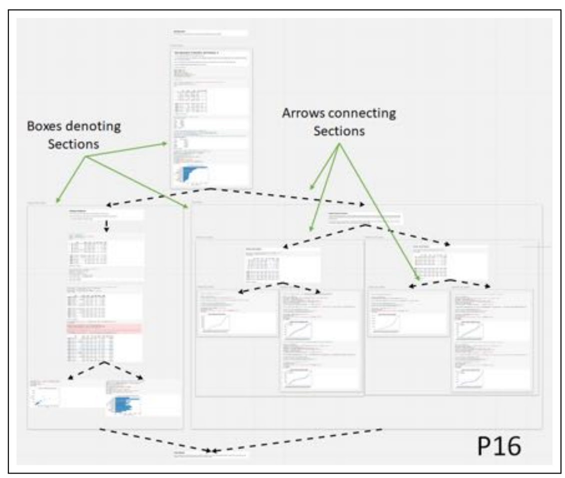

Fig. 5. Example Workboard approach, with arrows uses to represent branching flow as a DAG.

different data, same analysis, 15 layouts aligned similar charts for the Fairfax and Henrico data subsets horizontally, like in Figure 4 and 4 layouts aligned these charts vertically, like in Figure 3. Those who used the Multi-Column design pattern almost universally aligned the charts horizontally; the only layout who did not do this kept the analyses in one column.

These layouts may have utilized the space in this way to ease comparisons between Fairfax and Henrico data. For same data, different analysis, 4 layouts aligned the charts for all Virginia counties horizontally, like in Figure 3.

Feature Column	LINEAR							
Board ID	P02	P07	P10	P11	P12	P22	P23	SUM
columns	Y	Y	Y	Y	Y	Y	Y	7
rows	Y				Y		Y	3
scratch space				Y				1
boxes								
clusters								
sticky notes								
labels on arrows								
notebook outline								
arrows								
one-to-one arrows								
one-to-many-arrows								
many-to-one arrows								

Feature Column	MULTI-COLUMN								
Board ID	P01	P03	P04	P05	P08	P20	P21	P25	SUM
columns	Y	Y	Y	Y	Y	Y	Y	Y	8
rows	Y	Y	Y	Y	Y	Y	Y	Y	8
scratch space									
boxes									
clusters									
sticky notes								Y	1
labels on arrows									
notebook outline							Y		1
arrows			Y				Y	Y	3
one-to-one arrows			Y				Y	Y	3
one-to-many-arrows									
many-to-one arrows									

Feature Column	WORKBOARD										
Board ID	P06	P09	P13	P14	P15	P16	P17	P18	P19	P24	SUM
columns	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10
rows	Y	Y	Y	Y	Y		Y		Y	Y	8
scratch space			Y								1
boxes				Y	Y	Y	Y				4
clusters								Y	Y	Y	3
sticky notes				Y				Y		Y	3
labels on arrows								Y			1
notebook outline											
arrows	Y				Y	Y	Y	Y	Y		6
one-to-one arrows					Y	Y	Y	Y	Y		5
one-to-many-arrows	Y					Y	Y	Y	Y		5
many-to-one arrows	Y					Y			Y		3

Fig. 6. Low Level Features by Board ID

3) Low-Level Features: In addition to analyzing high-level strategies, we also noted use of certain low-level features, as seen in Figure 6. Common low-level features included columns of cells, rows of cells, and arrows to denote flow.

Beyond that, the authors noticed several features used for grouping related cells, such as boxes (or frames) around similar cells or spatial clustering of similar cells in the same general area, with clusters separated by white space. A few participants even used Miro's [9] virtual sticky notes feature to label different clustered sections or make notes to help others understand their layout. Finally, two participants appear to have left a few cells apart from the rest in 2D space; while this may be due to the participants deciding to stop after 40 minutes without finishing, it could also illustrate the use of empty 2D space as a form of scratch space or potential discard pile for cells that may not be deemed relevant.

In terms of run order strategies, we noted two approaches: explicit run order identification, and implicit run order identification. Explicit run order identification involves the use of arrows between cells to clearly note the run order, while implicit run order identification relied on intuitive rules, such as top-down execution of cells (or collections of cells) in a column and left-right execution in a row.

As for the use of arrows, we further delved into features present with those arrows. One participant labeled their arrows to denote the order in which multiple adjacent arrows were run. In addition, several different types of arrow usage were present. One-to-one arrows (8 instances) had one arrow come from a starting cell and going into an ending cell without any adjacent arrows going into the ending cell or coming out from the starting cell. One-to-many arrows (5 instances) had multiple arrows come from a starting cell and go to multiple different ending cells. Finally, many-to-one arrows (3 instances) had multiple arrows, each coming from different starting cells, go into one ending cell. There were no instances of many-to-many arrows.

B. Post-Task Survey Likert-Scale Results

Overall, participants rated the idea of 2D notebooks positively in comparison to their prior experience with 1D notebooks. All survey questions had at least a majority of votes on the positive side of the Likert scale. According to participants' survey responses, summarized in Figure 7, the tasks that are most likely to be facilitated by a 2D computational notebook are comparison activities, especially comparing results with different model parameters, different data, or different visualizations. However, some participants expressed skepticism about the usability of 2D computational notebooks for debugging, locating particular pieces of code, and performing analysis and development. Some participants doubted that navigation would be easier to perform in the 2D space.

Category	Question	Strongly Agree (3)	Agree (2)	Somewhat Agree (1)	Neither Agree Nor Disagree (0)	Somewhat Disagree (-1)	Disagree (-2)	Strongly Disagree (-3)	Mean	Median
High-Level Comparison of 2D to 1D	Better Info Layout	8	8	8	0	1	0	0	1.88	2.00
	Easier to Navigate	6	5	7	2	3	2	0	1.12	1.00
	More Beneficial	6	11	4	3	1	0	0	1.72	2.00
	Meaningful Non-1D Layouts	7	7	6	5	0	0	0	1.64	2.00
2D Layout Creation	Like Being Unrestricted	5	12	6	0	2	0	0	1.72	2.00
	Infinite Canvas Useful	6	12	4	1	2	0	0	1.76	2.00
	Used Multiple Areas	6	15	1	1	1	0	1	1.80	2.00
General Tasks in 2D vs. 1D	Collaboration	6	8	5	1	4	1	0	1.32	2.00
	Presentation	5	8	7	1	3	1	0	1.32	2.00
	Data Exploration & Preparation	5	10	3	3	3	1	0	1.32	2.00
	Analysis & Development	4	6	5	3	5	1	1	0.76	1.00
	Debugging	2	9	1	2	6	4	1	0.32	0.00
	Locating Code Fragments	4	6	6	3	4	2	0	0.88	1.00
	Branching Paths	8	8	4	3	2	0	0	1.68	2.00
	Sectioning Code	8	9	5	1	2	0	0	1.80	2.00
Comparative Analysis in 2D vs. 1D	Visualizations	12	7	6	0	0	0	0	2.24	2.00
	Different Parameters	13	8	2	2	0	0	0	2.28	3.00
	Different Data	13	6	4	2	0	0	0	2.20	3.00
Would Use	Use 2D as well as 1D	6	7	6	5	1	0	0	1.48	2.00

Fig. 7. Survey results with Likert-scale frequencies. Darker color indicates higher values.

C. Qualitative Survey Questions & Interview Results

We analyzed participants' qualitative survey and interview responses, and found several themes. We grouped these themes into several categories: 2D Notebook Benefits, 2D Notebook Challenges, 2D Notebook Features, and 2D Notebook Impressions. The first two focus on potential benefits of 2D Notebooks and design challenges for 2D Notebooks, respectively. 2D Notebook Features focuses on additional features that participants noted helped them make a compelling layout. Finally, 2D Notebook Impressions documents participants' attitudes towards the potential of 2D notebooks.

Table I shows our organization of themes by category and summarizes each theme with a sample participant quote. The table notes the number of participants whose comments expressed the same sentiment, either in a written qualitative survey response or interview format, as each theme.

V. Discussion

1D notebooks struggle with branching code paths, comparative analysis, and navigating longer notebooks. Due to these limitations and research into Space to Think, we sought to explore the potential of 2D Computational Notebooks to empower meaningful use of 2D space and to address certain issues of computational notebooks.

A. Organizing Notebooks in 2D

The linear split-cell approach, which can be done to some extent in Jupyter Notebooks using the Split-Cell extension, and the linear split-column approach, which is similar to Weinman et al.'s work on Fork it [15], shows how thoughtful

TABLE I
QUALITATIVE THEMES IN SURVEY & INTERVIEW RESULTS

Category	Theme	Sample Quote	Num. Responses
2D Notebook Benefits	Comparative Analysis	"I think the 2D notebook will be super useful for any kind of comparison analysis."	5
	Presentation	"This is a great tool for presenting data to layman audiences."	4
	Organization	"The 2D board is definitely hugely beneficial to organize code in a meaningful manner."	2
	Externalized Memory	"If I'm able to have everything listed in one document, I wouldn't have to be switching into web browser tabs to look at documentation or things like that."	1
	Collaboration	"2D space opens up the opportunity for multiple people working multiple spaces at the same time."	1
	Branching Code Paths	"I think 2D notebooks are better for the task with many branches."	1
2D Notebook Challenges	Tedious Navigation	"The zooming in and out and continuous scrolling to reorganize the tiles seemed tedious."	10
	Cognitive Load	"Organizing [notebook cells] might be a little tedious however, and needs to be planned in advance."	2
2D Notebook Features	Arrows	"The arrows were very useful, as they helped direct the flow of the narrative."	10
	Boxes or Frames	"The boxes also helped in grouping cells together."	7
	Table of Contents	"Main thing lacking is a table of contents created to link to the different sections."	1
	Run All In Column	"I would love to be able to "Run all the code in this column" or "Run all the code in the columns to the left/right"."	1
2D Notebook Impressions	Good	"I really enjoyed using this 2D computational notebook. It felt like a great and much easier way to reorganize some of my Jupyter notebooks!"	2
	Usability issues	"I think if the 2D Computational Notebook is developed in user-friendly and self-explanatory way, I would definitely choose to singly use a 2D Computational Notebook."	1
	Skeptical	"In terms of working with other programmers and developers, I feel that the linear layout would still work best."	2

modifications to 1D notebooks enable the use of 2D space within a traditional notebook environment. The prevalence of the Many-Column approach among participants, which can be argued to be an extension of the Split-Column Linear approach, suggests that a notebook extension to create multiple columns might be a feasible and effective way to enable 2D Space usage, like with the Split-Cell extension.

On the other hand, most users liked the flexibility of cell placement in the 2D computational notebook sketch using Miro [9]. In particular, participants that created Workboard layouts designed highly varied and creative layouts to express different aspects of the notebook structure. Given this, as well as the fact that all but one of the 21 2D layouts' run order could easily be intuited, a fully 2D Computational Notebook could be a valued tool, even in the presence of 1D notebooks with some limited 2D capabilities.

B. Enabling Strengths of 2D Notebooks

Practically all participants thought that the 2D Notebook environment would be useful for comparative tasks, such as comparing visualizations and comparing different parameters for model runs. They typically accomplished this by placing the cells or sequences of cells side-by-side, such as in the two aligned columns in Figure 4. Thus, 2D computational notebooks should be designed to enable users to compare results from different cells or even different branches of code in a flexible and easy-to-use way; in the context of multi-column designs, this might mean being able to change the order and alignment of the columns to better enable direct visual comparisons.

One interesting finding, based on the layouts and interviews, was the potential to run certain notebook segments in parallel. Users often specified these as parallel side-by-side branches in the cell layout, or with one-to-many arrows. This could be an excellent tool for data exploration and comparative analysis. For example, data analysts may want to try different machine learning models with a dataset for a classification task. By running adjacent columns of code in parallel, they can get results quicker and better compare model performance. They might then choose which path is best and mark it accordingly.

Finally, more sophisticated controls for running the cells in a 2D notebook may prove useful. Instead of just being able to run all cells or run a particular cell, a user may want to run a group of cells, such as those in a particular column or cluster. This would require being able to designate groups of cells to be operated on together. Some participants used explicit grouping features, such as boxes or columns or rows, to express such semantic grouping.

C. Addressing Tradeoffs of 2D Notebooks

While the 2D Computational Notebook concept shows promise, the results indicate some potential downsides.

One downside that some participants noted was that of tedious navigation. Given the participants' experiences, as detailed in the survey and interview responses, aiding navigation in intuitive, efficient ways seems a critical challenge for the design of 2D computational notebooks. Some suggestions for dealing with this issue include sectioning code cells using boxes or frames and enabling quick jumping to different sections, and enabling search utilities not only for keywords in code, but also in section and cell labels. In addition, a mini-map with interactive capabilities, such as being able to click on a cell and jump straight to it on the screen, may prove useful for easing navigation issues. One participant created an overview map that represented the flow of their notebook layout in miniature form. Finally, it may be useful to allow semantic zooming, where certain headers in markdown cells remain at a readable size even as a user zooms out.

Another potential downside is the possibility of increased cognitive load. When a user has to consider cell organization in addition to analysis, the additional cognitive load may prove burdensome. On the other hand, given the potential of notebooks to become "messy" [14] in 1D, it may be possible that considering cell organization during analysis may help avoid the problem of messiness to some extent.

Still, there are ways to address the possibility of increased cognitive load. First, a templating mechanism could be designed, in which users select from a set of common templates to pre-organize cells or provide initial structure. The templates should be based on our above observed high-level strategies. After selecting a template, the notebook might provide visual affordances such as pre-labeled sections in 2D space, and interactive affordances to fill in cells in the template. For example, a many-column template could provide a set of initial empty columns to fill in with cells.

Furthermore, AI methods could be developed to semi-automate cell organization through such actions as suggesting templates based on code structure and static analysis [25], semantic interaction [26], and active placement of cells. For example, an AI might recognize non-linear branches of 'same analysis with different data' and suggest a parallel multi-column template. Wenskovitch et al.'s work [25] on visualizing dependencies and relationships between computational notebook cells uses a dynamic graph structure that might be built upon to enable such methods.

D. Assessing Interest in 2D Notebooks

The results of the user study task provide evidence that, given the opportunity and reason to do so, users are willing to organize computational notebooks in 2D. The fact that, out of 25 participants, only 4 opted to align the cells in 1D (with one leaving a couple cells in "scratch space") supports this idea. With 19 out of 25 participants agreeing that they would add 2D Computational Notebooks to the set of analytic tools they use, the survey responses also support this assertion.

E. Limitations

This study has some limitations: the use of a completed notebook sketch, and the user study task-prompt wording.

1) Use of Completed Notebook Sketch: This study was performed on a sketch of a 2D Computational notebook in Miro [9] using images of cells from a 1D notebook, which makes it difficult to draw conclusions about how users would actually develop programs and narratives in a 2D computational notebook. The authors attempted to alleviate this issue by designing the user study prompt to get participants thinking about further development of the notebook. Still, having participants create the notebook from scratch instead of starting with all of its pieces could lead to different preferences in strategies and attitudes. However, this would require significantly more time and expertise from participants. We plan to explore this in future work.

2) User Study Task-Prompt Wording: When designing the study prompt, we considered how to hint that 2D layouts are an option without requiring 2D layouts. Our concern here was that, without any hinting, participants might default to the familiar 1D layout without any consideration of the possibilities of 2D layouts to address non-linear features. In reflection, we realize that the prompt could be read as strongly suggesting the creation of 2D layouts. Still, participants' survey responses explaining the reasoning behind their layouts largely suggest that participants found 2D space valuable for the notebook with non-linear features given to them.

VI. CONCLUSION

Computational notebooks have become a popular tool for data science and analysis; at their best, these notebooks enable crafting of meaningful, replicable computational narratives. However, the 1D nature of computational notebooks makes certain kinds of narratives harder to communicate and certain analytic tasks more difficult to perform. Thus, we sought to investigate the potential of 2D computational notebooks to address the non-linear narratives and improve upon the 1D computational notebook design.

Our work shows that 2D computational notebooks have potential and that users are interested in them as an additional tool. The ability to easily compare results, visually represent branching analyses, and present non-linear narrative structures in a semantic way seemed valuable to our participants. Participants' layouts were approximately evenly split between three major strategies: primarily linear, multi-column, and workboard strategies. The workboard strategy included directed graph layouts and complex nestings of other strategies. Approximately half of the participants also made use of additional annotation features, such as arrows, labels, and boxing.

However, 2D computational notebooks appear to have some potential difficulties for further research and innovation. The addition of a second dimension, while empowering more flexible placement of items, does complicate navigation. If 2D computational notebooks are to be successful, effectively aiding navigation appears to be a priority. In addition, given the increased complexity brought by organizing the layout of cells in 2D, layout aids such as templating may prove helpful in minimizing the effort necessary for organization during analysis. Still, it appears that 2D computational notebooks could very well provide a potent 'space to think' for data scientists.

REFERENCES

[1] J. Singer, "Notes on notebooks: Is jupyter the bringer of jollity?" in *Proceedings of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 2020, pp. 180–186.

[2] P. Jupyter. (2021) Project jupyter — home. [Online]. Available: https://jupyter.org/

[3] A. Rule, A. Tabard, and J. D. Hollan, "Exploration and explanation in computational notebooks," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–12.

[4] J. M. Perkel, "Why jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7732, pp. 145–147, 2018.

[5] F. Perez and B. E. Granger, "Project jupyter: Computational narratives as the engine of collaborative data science," *Retrieved September*, vol. 11, no. 207, p. 108, 2015.

[6] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, "Enterprise data analysis and visualization: An interview study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2917–2926, 2012.

[7] C. Andrews, A. Endert, and C. North, "Space to think: large high-resolution displays for sensemaking," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 55–64.

[8] L. Lisle, X. Chen, J. E. Gitre, C. North, and D. A. Bowman, "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, 2020, pp. 331–337.

[9] Miro. (2022) The visual collaboration platform for every team — miro. [Online]. Available: https://miro.com/index/

[10] D. E. Knuth, "Literate programming," *The computer journal*, vol. 27, no. 2, pp. 97–111, 1984.

[11] S. Chattopadhyay, I. Prasad, A. Z. Henley, A. Sarma, and T. Barik, "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*, 2020, pp. 1–12.

[12] J. Liu, N. Boukhelifa, and J. R. Eagan, "Understanding the role of alternatives in data analysis practices," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 66–76, 2019.

[13] M. B. Kery, M. Radensky, M. Arya, B. E. John, and B. A. Myers, "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*, 2018, pp. 1–11.

[14] A. Head, F. Hohman, T. Barik, S. M. Drucker, and R. DeLine, "Managing messes in computational notebooks," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[15] N. Weinman, S. M. Drucker, T. Barik, and R. DeLine, "Fork it: Supporting stateful alternatives in computational notebooks," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–12.

[16] M. B. Kery, A. Horvath, and B. A. Myers, "Variolite: Supporting exploratory programming by data scientists." in *CHI*, vol. 10, 2017, pp. 3 025 453–3 025 626.

[17] M. B. Kery and B. A. Myers, "Interactions for untangling messy history in a computational notebook," in *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2018, pp. 147–155.

[18] A. Rule, A. Birmingham, C. Zuniga, I. Altintas, S.-C. Huang, R. Knight, N. Moshiri, M. H. Nguyen, S. B. Rosenthal, F. Pérez *et al.*, "Ten simple rules for writing and sharing computational analyses in jupyter notebooks," 2019.

[19] R. Kosara and J. Mackinlay, "Storytelling: The next step for visualization," *Computer*, vol. 46, no. 5, pp. 44–50, 2013.

[20] N. Kirshenbaum, K. Davidson, J. Harden, C. North, D. Kobayashi, R. Theriot, R. S. Tabalba Jr, M. L. Rogers, M. Belcaid, A. T. Burks *et al.*, "Traces of time through space: Advantages of creating complex canvases in collaborative meetings," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. ISS, pp. 1–20, 2021.

[21] A. Bragdon, S. P. Reiss, R. Zeleznik, S. Karumuri, W. Cheung, J. Kaplan, C. Coleman, F. Adeputra, and J. J. LaViola Jr, "Code bubbles: rethinking the user interface paradigm of integrated development environments," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, 2010, pp. 455–464.

[22] A. Burks, L. Renambot, and A. Johnson, "Vissnippets: A web-based system for impromptu collaborative data exploration on large displays," in *Practice and Experience in Advanced Research Computing*, 2020, pp. 144–151.

[23] M. Konkiewicz. (2020) Jupyter notebook extensions (part 2). [Online]. Available: https://towardsdatascience.com/jupyter-notebook-extensions-part-2-55fdb2c38348

[24] R. Suzuki, T. Takahashi, K. Masuda, and I. Choh, "Implementing node-link interface into a block-based visual programming language," in *International Conference on Human-Computer Interaction*. Springer, 2018, pp. 455–465.

[25] J. Wenskovitch, J. Zhao, S. Carter, M. Cooper, and C. North, "Albireo: An interactive tool for visually summarizing computational notebook structure," in *2019 IEEE Visualization in Data Science (VDS)*. IEEE, 2019, pp. 1–10.

[26] A. Endert, P. Fiaux, and C. North, "Semantic interaction for sensemaking: inferring analytical reasoning for model steering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2879–2888, 2012.