

DreamSheets: Spreadsheets as Exploratory User Interface for Text-To-Image Models

*Frederick Kim
J.D. Zamfirescu-Pereira
Shm Almeda
Björn Hartmann, Ed.*



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-248

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-248.html>

December 1, 2023

Copyright © 2023, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

I would like to thank my research advisor, Professor Björn Hartmann, for his guidance on this work and support in my academic career. This work would not have been possible without the efforts of J.D. Zamfirescu-Pereira and Shm Garanganao Almeda, who are equal partners in the authorship of this work. Finally, I would like to thank my family for the boundless love and unwavering support they have given me.

DreamSheets: Spreadsheets as Exploratory User Interface for Text-To-Image Models

Kyu Won Kim

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for
the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee



Björn Hartmann
Research Advisor

April 25, 2023

(Date)

★ ★ ★ ★ ★ ★ ★



Kosa Goucher-Lambert
Second Reader

May 4, 2023

(Date)

Abstract

Users of Text-to-Image (TTI) models like DALL•E and Stable Diffusion typically engage in a lot of iteration, exploring a design space, to achieve satisfactory outcomes. This design space’s input parameters consist of (1) prompt text spanning image content and style, and (2) stochastic (e.g., random seeds) and/or other opaque (e.g., classifier-free guidance) variables. Spreadsheets offer a natural interface for end-users to engage in design space exploration and rapid iteration in a “what-if” style. In this work, we present **DREAMSHEETS**, a spreadsheet interface for creating images using TTI models. **DREAMSHEETS** enables exploration of multiple input changes simultaneously, affording prompt-crafting using spreadsheet formula construction. Crucially, we also introduce a set of new functions that enable rapid exploration of the prompt input space, utilizing GPT-3 to generate context-relevant lists of prompt keyword options, new variations on existing prompts, and more. These functions enable **DREAMSHEETS** users to rapidly explore the neighborhood around an initial prompt, leveraging the spreadsheet’s simultaneous display of these prompt-adjacent images. In a small formative study, we explore how the spreadsheet metaphor and these new functions impact participants in achieving and understanding artistic goals, concluding with some lessons learned for future designers of exploratory TTI-based systems.

Acknowledgement

I would like to thank my research advisor, Professor Björn Hartmann, for his guidance on this work and support in my academic career. This work would not have been possible without the efforts of J.D. Zamfirescu-Pereira and Shm Garanganao Almeda, who are equal partners in the authorship of this work. Finally, I would like to thank my family for the boundless love and unwavering support they have given me.

Contents

1.	Introduction	6
2.	DREAMSHEETS Usage Scenario	8
3.	Background & Related Work	10
3.1.	Creativity Support	10
3.2.	Prompting & Text-to-Image Model Workflows	10
3.3.	Design Space Exploration of Images	11
3.4.	Alternative Modes of Organizing Computation	12
4.	DREAMSHEETS Design & Implementation	12
4.1.	Design	13
4.2.	Implementation	13
5.	User Study Design	15
5.1.	Participants	15
5.2.	Task	15
5.3.	Data Analysis	18
6.	Findings	19
6.1.	DREAMSHEETS In Use	19
6.2.	Challenges	22
6.3.	Measuring Satisfaction and Usage	22
7.	Discussion	24
7.1.	Learning Curves	24
7.2.	Integration Trade-Offs	25
7.3.	Creativity Support	25
7.4.	Lessons for Future Tool Designers	26
7.5.	Limitations and Future Work	27
8.	Conclusion	28
	Bibliography	29

Appendices **34**

- A. Likert Questions 34
- B. Tutorial Screenshots 35
 - B.1. TTI Introduction Sheet 35
 - B.2. GPT Functions' Introduction Sheet 36










	A	B	C	D	E	F	G
1				14770	66111	17154	7543
2		City in ruins. Post-apocalyptic, crumbling buildings.	=ALTERNATIVES(B2) City in ruins, view from a broken window. Burnt sky.	=IMAGE(TTI(\$C2 , D\$1))			
3			Post-apocalyptic highway, vehicle husks. Burnt sky.				
4		Post-apocalyptic scene objects	City behind walls, sewer draining into ocean.				
5		=GPT_LIST(B4) abandoned car	=B5 & " in a " & C2 abandoned car in a city [...]	=IMAGE(TTI(\$C5 , D\$1))			
6		torn billboard	torn billboard in a city [...]				
7		broken streetlight	broken streetlight in a city [...]				
8		rubble pile	rubble pile in a city [...]				

Figure 1.: DREAMSHEETS in typical use. Gray lozenges show user-entered formulas, while dashed lines show cell values containing computed or generated content. Here, an initial prompt (B2) is expanded using =VARIATIONS into column (C2:C4). Together with a row of seeds (D1:G1), this yields a set of images (D2:G4). Further exploration is enabled by =GPT_LIST (B5:B8) and concatenation ($\&$) into new prompts (C5:C8).

1. Introduction

Artists and other users of new Text-to-Image (TTI) models like DALL•E and Stable Diffusion are faced with a cacophony of tools and models with which to create images from text. Despite their variety, these tools typically offer a very straightforward interface: a prompt input, perhaps also a negative prompt input, sliders for hyperparameters, and an output area showing a few output images and some saving options. The *process* that users engage in when using prompt-input models, however, is **highly iterative and dynamic**, involving substantial editing and re-editing of prompts and evaluating model outputs [16, 45]. For TTI models, emergent workflows also involve resources beyond the generators themselves: prompt guidebooks [28], image editing software from MS Paint to Photoshop, and a variety of web- and app-based front-end interfaces (see, e.g., [34], for an example).

Because the possible input space for these models is opaque [1] and massive—comprising prompts, seeds, hyperparameters, and even other images as inputs—users are forced to engage in a challenging search process through this space. Commonly used commercial and open-source tools offer relatively little support around prompt construction and visualizing this space beyond producing single or small batches of images for a specific point in input space (or neighboring points with different seeds, in the case of batches), but these models are so useful that users engage in *ad hoc* design exploration despite these limitations [16]. Prior research has shown that design space exploration can be effectively aided by considering many alternatives in parallel [13, 30], tracking exploration history [14, 17], offering suggestions for possible input shifts, and effective organization [22, 32, 40], capabilities lacking in these commonly-used tools.

In this paper, we present *DREAMSHEETS*, a creativity support tool for image generation using text-to-image models, based on the spreadsheet metaphor. *DREAMSHEETS* allows users to rapidly generate prompts and variations on prompts, and explore generated images outputs using a familiar grid layout and a “what-if” interaction model that can easily track history and supports many different styles of knowledge organization and management.

Prompt design is a challenge in its own right—the modifications to a prompt necessary to steer a model output toward a desired outcome is not always intuitive, and can be challenging to find. To aid prompt exploration, *DREAMSHEETS* also introduces a new set of LLM-based functions (*functions* are often called “operators” or “formulas” in the spreadsheet context) for manipulating prompts directly, including `GPT_LIST` and `LIST_COMPLETION` for generating or extending a list of items of a certain description, `EMBELLISH` to add detail, and `ALTERNATIVES` to generate multiple variations of a seed prompt (see Table 1 for a full list). These features are all embedded within an actual spreadsheet (built on Google Sheets) that recomputes and rerenders images, allowing drag-based “autofill” and other common spreadsheet functionality.

Finally, using *DREAMSHEETS*, we run a small exploratory study to better understand how a spreadsheet interface can aid or hinder the use of TTIs and find that while the spreadsheet metaphor and *DREAMSHEETS*’ functions do assist in rapid exploration of the input space, users varied substantially in their ability and willingness to rapidly iterate. The low-structure nature of the spreadsheet environment could make “getting started” challenging, and also forced users to fit their exploratory process into rows and columns – which not all users find naturally intuitive or appealing to their creative processes.

We make 3 contributions through this work:

1. DREAMSHEETS: a tool that provides a spreadsheet interface on top of Text-to-Image (TTI) and Large Language Models (LLMs).
2. A rich exploratory evaluation of how a spreadsheet interface supports and hinders end users in exploring the TTI generation design space.
3. A set of implications for the design of future TTI and LLM tools.

Function Name	Description
<code>TTI(prompt, [seed], [cfg])</code>	Generate an image (and return a URL pointing to it) using the given <i>prompt</i> and, optionally, <i>seed</i> and a classifier-free guidance (<i>cfg</i>) parameters.
<code>GPT(prompt)</code>	LLM function for arbitrary <i>prompt</i>
<code>GPT_LIST(prompt, [length]=5, [transpose]=false)</code>	Populates <i>length</i> cells in a row (<i>transpose</i> = TRUE) or column (<i>transpose</i> = FALSE) with words/phrases of type <i>prompt</i> .
<code>LIST_COMPLETION(prompt, [length], [transpose])</code>	Like <code>GPT_LIST</code> , but <i>prompt</i> is a list of items, rather than a class description.
<code>SYNONYMS(prompt, [length], [transpose])</code>	Generates a list of synonyms to <i>prompt</i> .
<code>ANTONYMS(prompt, [length], [transpose])</code>	Generates a list of antonyms to <i>prompt</i> .
<code>DIVERGENTS(prompt, [length], [transpose])</code>	Generates a list of “divergent” words to <i>prompt</i> .
<code>ALTERNATIVES(prompt, [length], [transpose])</code>	Generates a list of alternative wordings for <i>prompt</i> .
<code>EMBELLISH(prompt)</code>	Generates an embellished alternative to <i>prompt</i> , commonly using more specific or detailed words.

Table 1.: The LLM-based functions available in DREAMSHEETS. Each list-producing function additionally has an `_T` alternative (e.g., `SYNONYMS_T`) that calls the original function with `transpose = TRUE`.

2. DREAMSHEETS Usage Scenario

We introduce a running scenario to illustrate DREAMSHEETS’s functionality.

Paula T. Artist works for a medium-size movie studio. She wants to create some concept art for an upcoming film she plans to pitch about a dystopian vision of the Pacific Northwest region within the United States. Inspired by an initial image of a post-apocalyptic cityscape (Fig. 1, cell B3), Paula wants to generate a few complementary images to use in storyboarding her film. She's more interested in the aesthetic feel of these images, wanting to convey a certain mood, rather than any specific content—in others words, there is not a specific image Paula has in mind that she is trying to produce.

Paula begins by brainstorming ideas for different scenes in the movie. Based on her ideas, she creates a written description that she can use as a prompt with DREAMSHEETS's TTI() function: City in ruins. Post-apocalyptic, crumbling buildings. She uses a cell reference (e.g., B2) to convert her text prompt into an image URL that's then displayed directly inside a cell using the built-in IMAGE() function. Because DREAMSHEETS is built on top of a functional spreadsheet (Google Sheets), cells automatically update in response to changes in referenced cells—thus changes to a prompt in one cell are rapidly reflected in changes to any images referencing that cell.

Dissatisfied with initial output, Paula turns to DREAMSHEETS's built-in LLM-based ALTERNATIVES function (Fig. 1, cells C2:C4—and see Table 1, too) to generate new prompts: "City in ruins, view from a broken window. Burnt sky.", "Post-apocalyptic highway, vehicle husks. Burnt sky.", and "City behind walls, sewer draining into ocean." Using a row of random seeds, 14770, 66111, 17154, and 7543 (Fig. 1, cells D1:G1), Paula quickly generates a 2D grid of images based on the cross-product of these prompts and seeds (Fig. 1, cells D2:G4). She does so by first entering the formula =IMAGE(TTI(\$C2, D\$1)), and then dragging the "autofill" handle to fill a 2D region of 12 cells to span the three generated prompts and four seed values.

In a first pass at refining these images, Paula iterates on her prompts a few times to see what variations result, adding a new row for each prompt variation in order to preserve her exploration history.

Turning to another of DREAMSHEETS' LLM-based prompt exploration functions, GPT_LIST (Fig. 1, cells B5:B8), Paula explores adding a few post-apocalyptic objects to one of her earlier prompts (Fig. 1, cells C5:C8) using string concatenation (&). Calling GPT_LIST with a reference to the string "Post-apocalyptic scene objects" returns the values abandoned car, torn billboard, broken streetlight, and rubble pile; using the formula =B5 & " IN A " & C2, Paula programmatically generates new prompts by combining these objects with the prior prompt, e.g., abandoned car in a City in ruins, view from a broken window. Burnt sky.

Iterating in this fashion, Paula tries about ten or twenty prompts over the course of about as many minutes, ultimately finding a set of images that she feels effectively captures her desired mood and is aesthetically consistent with one another.

As a major side benefit, when Paula is satisfied with her generated images and turns to sharing her work with collaborators, the underlying Google Sheet that Paula used can easily be shared. The embedding of her explorations into a collaborative online spreadsheet means that Paula’s collaborators can also explore the full history of how Paula came to the prompts and images she ultimately selected, and what images and prompts she considered along the way.

3. Background & Related Work

We relate our work to the larger topics of Creativity Support tools and Generative AI workflows; and to prior work on design space exploration tools and ways to organize computational work.

3.1. Creativity Support

In 2007, Shneiderman identified four underlying design principles for creativity support tools (CSTs): *support exploratory search, enable collaboration, provide rich history-keeping, and design with low thresholds, high ceilings, and wide walls* [37]. A more recent body of research explores how CST design can aid users’ creative processes and productivity [8, 9]. Spreadsheets are themselves an example of a tool that supports creative exploration, enabling users to separate fixed values from values they want to vary, affording effective exploration and evaluation of “what-if” scenarios [31, 37].

3.2. Prompting & Text-to-Image Model Workflows

The rapid emergence of the “pre-train, prompt, predict” paradigm in NLP [19] has led to an explosion of generative models steered by user input of text-based prompts, including text generation models like GPT-3 [4], as well as image generation models like DALL•E [29] and Stable Diffusion [33]. Text-to-Image systems in particular allow designers and other creatives to use visual concepts expressed as text prompts to generate images in a wide variety of subjects and styles, allowing artists to explore a design space and discover novel sources of inspiration.

Though prompting can appear as easy as instructing a human, crafting effective prompts can be a challenge [20, 45], and how a prompt directly impacts model outputs is an active area of NLP research [19, 35]. Choosing the right language to achieve desirable visual results in these

prompt-based interactions can be difficult, presenting challenges for users seeking to reliably incorporate TTI systems into their exploratory workflows and accomplish their creative goals [45]; this has motivated online user communities [2] and researchers to develop and investigate new prompting techniques and tools [10, 20, 21].

3.3. Design Space Exploration of Images

An early typology of different approaches to design space exploration can be found in Woodbury and Burrow [40]. When visual judgment of the produced artifact by a human designer is the primary method of evaluation, as it is in computer graphics and animation, prior work has often focused on browsing interfaces, such as in Marks et al.’s seminal Design Galleries [23].

Various tools use information visualization techniques such as scatterplots [24, 27] or histograms [11] of explored designs, or parallel coordinates [26] that plot multiple designs on a set of shared axes. Interaction techniques for browsing include multi-step galleries [23]; map metaphors [38]; or faceted browsing [11].

From the explored designs, users may wish to pursue multiple alternative options. Some interfaces thus support operating on user-defined design alternatives. The number of pursued alternatives is usually orders of magnitude smaller than the number of algorithmically explored designs – e.g., in GEM-NI [44], Subjunctive Interfaces [22], and Juxtapose [13].

Exploration tools for computationally expensive design spaces typically precompute a large number of samples. These tools must consider how to handle user requests to view points in the design space that were not precomputed: some forbid it [23], while others interpolate between precomputed points using computationally inexpensive proxy models [36]. In DREAMSHEETS, parallelization and a cloud API allow simultaneous generation of output images, avoiding the need for precomputation or lengthy waits.

Spreadsheets’ usefulness for visual design space exploration in part stems from the intrinsic 2D matrix layout enabling “small multiples”, a term Edward Tufte popularized [39] as an answer to the question “compared to what?” In a 2D matrix, a large number of images can be readily compared with each other, and the best candidate images identified. Used in this way, spreadsheets additionally have a rich history of serving as vehicles for other, non-accounting exploratory work, as far back as 1994 for information visualization of data and images themselves [7, 18].

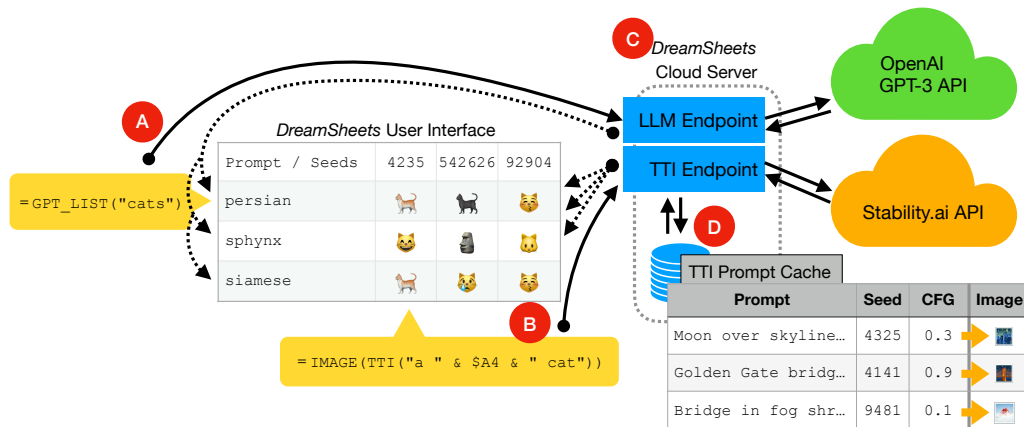


Figure 2.: The DREAMSHEETS implementation. LLM (A) and TTI (B) functions fetch from separate endpoints of the DREAMSHEETS cloud server (C) which forwards requests to the OpenAI ChatGPT and Stability.ai Stable Diffusion cloud-based APIs. TTI requests are cached (D) using a hash of (prompt text, seed, classifier-free guidance) as the key.

3.4. Alternative Modes of Organizing Computation

In considering other paradigms for helping image creators explore the design space of prompts, seeds, and other generative model hyperparameters, we considered but ultimately rejected a number of alternatives: computational notebooks like Jupyter¹, node-based visual languages like Unreal Engine’s Blueprint Visual Scripting², as well as Promptchainer [41] and other recent work with LLMs [10, 42], including early attempts at extending the spreadsheet paradigm [5].

4. DREAMSHEETS Design & Implementation

Our primary design goal was to enable design space exploration of TTI model inputs and outputs, using an approachable computational approach, in service of two user outcomes: (1) achieving a desired generated image; and (2) developing a better understanding of model capabilities and how specific input prompt text and other hyperparameters correlate with specific outputs.

We ultimately chose a spreadsheet metaphor for a few reasons: first, its ubiquity—any new paradigm would require substantial upfront training, removing a major potential benefit. Second, spreadsheets avoid a requirement for *users* to constructs a spatial organization—not only do

¹<https://jupyter.org/>

²<https://docs.unrealengine.com/en-US/Engine/Blueprints>

spreadsheets support this organization out-of-the-box, they, to some extent, also *enforce* this organization. And third, any approach requiring writing traditional code (Python, JavaScript, etc.) would dramatically raise the bar for background knowledge required to use any resulting tool.

4.1. Design

From a design perspective, integrating TTI model functionality was relatively straightforward—but we devoted significant design energy to developing a set of prompt exploration functions (see Table 1), themselves based on prompting a LLM.

Drawing on prior work on prompt design, we identified the testing of alternative phrasings and the addition of detail as core activities in TTI prompt exploration [20, 28]. These activities help users explore neighboring points in the design space; those points in turn can suggest fruitful *directions for further changes* to prompts. We operationalized support for these activities as the ALTERNATIVES, DIVERGENTS, and EMBELLISH functions. Similarly, synonym and antonym generation are core NLP building blocks, useful for creating variation by targeting specific words within a longer prompt—we integrated these capabilities directly as well through the SYNONYMS and ANTONYMS functions.

Critically, to use these concepts in a spreadsheet paradigm where they can support the generation of *sets* of images, we wrote these functions to output *lists* of values that can populate across a column (or row) of cells. Once in a row or column of cells, these variations can be referenced by cell in traditional spreadsheet style, and used as prompt building blocks (or as prompts directly) to generate new images.

The fundamental concept here of a *list of prompts* or prompt components also opens the door to functions that *extend these lists*, allowing users to build on a conceptual list by using a *seed list*, rather than requiring a declarative description of the items within the list.

4.2. Implementation

We explored different services to provide the underlying spreadsheet functionality for DREAMSHEETS, including building our own spreadsheet interface from scratch, open source spreadsheets HandsOnTable³ and LuckySheet⁴, and both Excel and Google Sheets.

One major challenge in integrating with an existing spreadsheet is the relatively long latency of image generation itself: up to 15 seconds or more, even when using cloud APIs. Spreadsheet

³<https://handsontable.com>

⁴<https://github.com/dream-num/Luckysheet>

users are accustomed to rapid updates and recomputations in response to changes in cell values—a multi-minute delay resulting from a backlog of image prompt updates and, consequently, new image generations, would have been unacceptably slow. This need drove our use of the Stability.ai API⁵, which supports parallel image generation requests and offers sub-15 second response times. These were critical features to enable full-scale “small multiples” visualizations from combining a row of one set of input features (e.g., prompt text) with a column of another such set (e.g., seeds).

Ultimately, we selected Google Sheets as the spreadsheet interface, as it is easily extensible and accessible to most people. Google Sheets’ Apps Script environment lets developers create add-ons in a JavaScript-like environment, has a sufficiently long timeout (30 seconds) for custom functions, and allows users to continue to edit the sheet even while our custom functions, which required back-end calls to TTIs and LLMs, awaited responses.

As a side benefit, because Google Sheets is already an online-native platform, rapid collaboration is built-in and the spreadsheets can be easily shared.

We implemented DREAMSHEETS as a Google Sheets Apps Script add-on and a proxy web server written in Javascript and ExpressJS. The add-on adds custom functions described in Table 1, making the corresponding requests to the proxy web server which handles caching and calling the appropriate API to either Stability.ai or OpenAI. Figure 2 illustrates how the proxy server facilitates communication between the Google Sheets add-on and Stability.ai or OpenAI. For the TTI function, the proxy server makes a hash using a combination of the prompt, seed, and guidance values and checks if the image has been generated before. Otherwise, an API call is made to Stability.ai to generate a 512×512 image which is then cached in the file system for easy retrieval in the future.

The LLM-based functions that return a list utilize OpenAI’s ChatGPT with gpt-3.5-turbo. To ensure that ChatGPT returns a properly formatted list with the appropriate length, it is initialized with the following messages:

```
system: Respond with a Javascript array literal with the given length in parentheses
```

```
user: types of animals (length: 5)
```

```
assistant: ["dog", "cat", "frog", "horse", "deer"]
```

```
user: [PROMPT] (length: [LENGTH])
```

The prompt for the LLM-based functions modified the user inputted text by prepending additional instructions according to each function’s description prior to sending it to the proxy server. The complete prompts used are listed below:

⁵<https://api.stability.ai/docs>

- LIST_COMPLETION: ‘Similar items to this list without repeating “[LIST]”’
- SYNONYMS: ‘Synonyms of “[PROMPT]”’
- ANTONYMS: ‘Antonyms of “[PROMPT]”’
- DIVERGENTS: ‘Divergent words to “[PROMPT]”’
- ALTERNATIVES: ‘Alternative ways to say “[PROMPT]”’
- EMBELLISH: ‘Embellish this sentence: [PROMPT]’

A critical consideration for the implementation of DREAMSHEETS was error handling. Since LLMs are not strictly bound to return text in a parsable format, handling errors with clear messaging was necessary. The option for the user to manually rerun the function was included in the event that the LLM returned an unparseable result.

5. User Study Design

We performed a user study of DREAMSHEETS with 12 participants to better understand how participants might leverage the functionality of DREAMSHEETS for a text-to-image generation workflow similar to that described in the Usage Scenario.

5.1. Participants

We recruited participants through email lists and social media, advertising it as a study of a novel “Stable Diffusion Support Tool” involving spreadsheets. Each interview took place through a Zoom call that lasted approximately 60 minutes; participants were compensated \$20. We invited interested participants to sign up via a short pre-screening survey, through which they also reported their prior experience with spreadsheets and with TTI models. Table 2 lists our participants’ demographic details and prior experience; all 12 reported some spreadsheet experience, with most (10 out of 12) reporting frequent use (many times or daily). 10 out of 12 participants also had some experience with TTI models, and only 1 participant (P1) reported no prior experience with LLMs.

5.2. Task

We had a few requirements in mind when selecting a task for our study: (1) it should not *require* specific content within the generated images, like a particular person, object, or place—DREAMSHEETS

ID	Spreadsheet Use	TTI Model Experience	LLM Experience	Art Background
P1	Occasional	Limited experimentation	None	Casual
P2	Daily	None	ChatGPT, Github Copilot	None
P3	Often	Limited experimentation	ChatGPT, many others	None
P4	Often	Limited experimentation	ChatGPT, many others	None
P5	Often	Substantial; understands capabilities	ChatGPT, many others	None
P6	Often	None	ChatGPT	Casual
P7	Often	Substantial; understands capabilities	ChatGPT, several others	Amateur
P8	Occasional	None	ChatGPT	None
P9	Often	Extensive TTI & prompt experience	ChatGPT, several others	None
P10	Often	Limited experimentation	ChatGPT	Casual
P11	Daily	Extensive TTI & prompt experience	ChatGPT, Bing Chat	Freelance/casual
P12	Often	Extensive TTI & prompt experience	ChatGPT, many others	Casual

Table 2.: Data on our 12 participants' prior experiences, if any, with spreadsheets, text-to-image (TTI) models and tools, Large Language Models (LLMs), and art background, as each participant self-reported in questionnaires completed at the start of the study.

does not have any image manipulation or image-input capability, and common TTI workflows often rely on image manipulation to incorporate specific content; (2) it should encourage exploration along both content and style axes; and (3) it should be achievable within a reasonable amount of time.

We operationalized these requirements by designing a *concept art* creation task, requiring no specific content or style, and we developed a scenario (see §2) in which participants use a single initial image for inspiration for set of three new images that fit within the broader arc of an unspecified narrative.

From there, our interview protocol proceeded in two steps: a brief tutorial to DREAMSHEETS and its functionality, followed by an observation of participants engaged in the task and encouraged to think aloud.

Tutorial

After briefly discussing the participants’ prior experiences with spreadsheets and AI generation tools, we introduced DREAMSHEETS and provided a tutorial in multiple phases. First, participants were briefly reminded of general spreadsheet functions, including using cell references, the \$ symbol (to specify absolute references), and dragging to expand a formula through “autofill” into a longer row, column, or matrix of adjacent cells.

Next, we introduced participants to the IMAGE and TTI functions. We used an example sheet (see Appendix B.1) to walk participants through creating a prompt in one cell, then reference it in a function call contained in another cell, to preserve the prompt in plain text form while showing the generated image in a neighboring cell. We also demonstrated how cell contents, including images, change when referenced cells themselves change. A 2-D matrix of generated images with incrementing CFG values along the rows and varying seed values along the columns demonstrated creating a cross product of two variables and observing their differences.

We gave participants about 5-10 minutes to try these TTI functions and become accustomed to their use.

Next, we introduced the LLM-based functions available for use in DREAMSHEETS (listed in Table 1), using another example sheet (see Appendix B.2). We showed, through examples, how to use spreadsheet interactions to utilize these text-generation and modification functions, and how they might be helpful prompt-engineering aids. We demonstrated how to incorporate references to other cells, concatenate results with other generated values (or manually written values), and expand cells across a row or column to create new prompts in parallel. Motivated by the

nondeterministic nature of the LLM-based function outputs, we also showed users how to copy “values,” rather than functions, to interrupt the nondeterminism and enable more robust workflows.

We gave participants another 5-10 minutes to try these LLM functions in conjunction with TTI.

Following this introduction, we asked participants to share their screens with us and “think aloud” so we could observe their workflow. Once users were comfortable with the TTI and GPT tools, we introduced the activity.

Activity

Next, we prompted participants with our task, inspired by the job faced by Paula T. Artist in our Usage Scenario (see §2):

Imagine that you are a movie director creating a new film. You wish to visualize how the movie might look, so you create the scene shown below. You want to explore how other parts of the movie might look like. Use this image as inspiration to create three additional reference shots with this tool. Using the GPT functions and TTI, explore different prompts in order to select the best reference image for your shots.

We included this prompt in the spreadsheet, as well as an image of a post-apocalyptic, ruined Seattle, complete with the Space Needle⁶ (see Fig. 1, cell B3).

We asked participants to start by coming up with a general idea of what the three shots should look like. We also asked participants to keep their prompts and the generated images in separate cells, and demonstrated how they could do this by entering the text prompt in one cell, then generating the image in a separate cell using a dynamic reference to the cell containing the text prompt in their TTI function call. This allowed us, and the participants, to see both the text prompt and its corresponding image explicitly displayed in parallel in the sheet, and as a side effect, further modeled passing cell references to the TTI function.

Lastly, participants were encouraged to think aloud as they worked through the task.

5.3. Data Analysis

Each interview was recorded and transcribed; we then engaged in an exploratory data analysis, observing where participants did and did not elect to make use of specific affordances of DREAMSHEETS, and how that use aided or hindered their goals. These observations were then aggregated across participants by categorizing their behaviors and comments using an affinity

⁶This image was borrowed with gratitude from Andy Salerno, whose blog post [34] originally inspired this work.

diagram [25] and documenting the specific actions that participants were observed taking through a limited service blueprint [3]. These HCI and service design methods suit our goal of discovering opportunities created by a new technology—instead of understanding how *existing* users engage in *established* work practices; there, a grounded theory approach could be more appropriate.

In addition to the qualitative data above, we also recorded two sets of quantitative data: first, we tracked usage data for DREAMSHEETS’s custom spreadsheet functions to help us understand which functions were most useful; second, we asked users to participate in a post-interview survey, consisting of seven questions ranging from ease of use to transparency and expressiveness, answered using a 5-point Likert scale.

6. Findings

In this section, we detail our observations of participant behavior in our user study, as well as details from our post-study survey and instrumentation.

Every participant was able to use DREAMSHEETS to generate and select at least one image they deemed a reasonable end-result, and there was wide variety across participants with respect to the DREAMSHEETS functionality used.

6.1. DREAMSHEETS In Use

DREAMSHEETS’s functions support certain behaviors related to our participants’ exploratory goals, and in practice these behaviors seemed to fit into these 3 broad categories:

1. Whole-prompt variation (via `EMBELLISH`)
2. Word-level prompt variation (via the LLM functions that generate words or short phrases, and concatenation)
3. Image-generation variation (via `TTI seed` and `cfg` parameters)

We also observed manual prompt authoring, without using any additional functions for assistance.

People who had trouble coming up with prompt language appreciated having the LLM-based functions.

A common starting strategy was devising an initial prompt manually, then using `EMBELLISH` to improve the prompt. For example, P4 authored the following for the first shot:

A vibrant futuristic Seattle cityscape lush with many happy people. Things seem almost too perfect...

Using EMBELLISH, P4 ultimately settled on the following prompt:

The sun shines brightly over the picturesque Seattle cityscape, with its vibrant, futuristic buildings and lush greenery. Everywhere, happy people laugh and chatter, and the atmosphere is one of contentment and joy. It almost seems too perfect to be real.

P4 felt that the embellished prompt would be “better” than their original, saying “I thought that [EMBELLISH] was the most useful for me because I know that the Stable Diffusion prompts should be overly specific.”

P5 liked that EMBELLISH returned a block of text in “one step,” in contrast to the other LLM-based functions that return individual words or short phrases, and then require concatenation or other work to generate full prompts.

The participants who felt most comfortable with string concatenation often used the LLM-based functions to introduce *word-level variation* to a manually authored prompt or prompt structure.

Participants with less prompt-engineering experience wrote their prompt in “English” and explored different options for the adjectives, verbs, or settings latent in their authored structure.

For example, P8 started with “humans hiding in a ruined city” and split these terms into separate cells to enable individual use of cell references and concatenation. They then used ALTERNATIVES(“ruined city”) and DIVERGENTS_T(“hiding”) to create a matrix of prompt variations including “humans masking in a destroyed metropolis” and “humans veiling in a devastated urban center.”

Though authored directly by participants, it is worth noting that these workflows may have been inspired by the examples provided during the tutorial phase, which created variations of “dogs on a hill” that follow the structure “[*type of animal*] on a [*landscape*]”.

Using LLM-based functions to get artistic variations was also observed. P1 wanted to make sure the generated images matched the style of the inspiration image, so they used SYNONYMS(“dark rust colored”) to see similar colors. P3 similarly ran SYNONYMS(“red”) to get different types of red colors.

Participants in general tended to be pleasantly surprised by even very unexpected or “weird” results. P5 struggled to generate an image that aligned with their creative vision of a spaceship in the sky, but ended up satisfied with an intermediate result.

Yeah, for some reason, I’m unable to get the spaceship at all... but I actually like this.

-P5

Using DREAMSHEETS to explore model capabilities Participants with more prompt-engineering experience also used the LLM-based functions for a more targeted approach to generating variation in their prompts. For example, P₁₁ generated a list of artists, then used DREAMSHEETS to test which ones Stable Diffusion “knew”—that is, which names had an impact on the generated image style that aligned with what the participant themselves knew about the artist’s style.

Does Stable Diffusion know the same artists I do? (P₁₁)

P₉, who also had extensive prompt-engineering experience, generated a list of filming angles and art movements, then created an image matrix using these to create variation along two axes and discover which “modifiers” influenced the image generation in a desirable way. They then selected terms to manually add to their prompt, which was structured as a comma-separated list of selected words and phrases.

P₉ identified user understanding of model “knowledge” as a common challenge that is part of their process in improving a prompt and in improving their effectiveness as a prompt engineer:

That’s always one of the challenges, when you’re trying to, like, understand what the AI ‘*knows*’ ...being able to get like a big enough data set enough, or enough images and enough seed variations to have some idea of what the AI’s ‘*idea*’ is ...What does it ‘*know*’ about this thing? (P₉)

P₉ and P₁₁ appeared to use DREAMSHEETS as part of a *targeted* TTI model creative *research* process. By testing Stable Diffusion’s vocabulary to improve their understanding of the model’s capabilities and quirks, they would develop an intuition for more effective prompts.

Even participants with less experience engineering prompts could be observed beginning to develop a working understanding of what the model “knew” throughout their creative process.

After dynamically adding “Unreal Engine” to a series of prompts and observing the impact on the regenerated images, P₁₂ exclaims:

Okay, let’s just add ‘Unreal Engine’...So it understood that! (P₁₂)

P₆, despite minimal experience with TTI and LLMs, also recognized this value:

I think it’s good to know how things change depending on different variables... I think the spreadsheet helps with navigating what exactly is changing within the image. (P₆)

Once participants had prompts or generated images they were generally satisfied with, they could use TTI hyperparameter variation (*seeds* and *cfg*) to get hyperparameter-level variations on the same prompts.

6.2. Challenges

Participants who felt they knew exactly what they wanted needed to be encouraged to explore further — they seemed hesitant to use the provided LLM functions, which inherently create new variations for them to consider.

Patience for wrangling spreadsheet formulas was limited: a number of participants tried only once or twice to get formulas or concatenation to work before giving up (P6, P8).

Some participants were also reluctant to put in the effort required to generate new options when they had already converged on an ideas they felt satisfied with.

It was easier if I just typed down different things, different sentences, and then generated [images with] that instead of like, randomly generating objects, since I had a specific image in mind. (P6)

P9 in particular found the *enforced* 2D matrix structure to be both helpful and a hindrance:

I like how flexible it is, but I dislike how flexible it is. It's powerful that I have all the power of a spreadsheet and all the flexibility that it affords. At the same time, it is a little bit cumbersome... to setup. I could see it being a bit of a bottleneck for someone who's less familiar with spreadsheets overall. (P9)

P3, P5, and P7 all struggled to make a spaceship "appear" - P9 was able to achieve this goal somewhat easily, perhaps drawing on their substantial prior prompt-engineering experience. P7 became particularly frustrated when, despite several attempts at creating targeted iterations of their prompt, they could not generate images featuring a spaceship, nor an orange sky, stating in exasperation that the model seemed to have a "*misunderstanding of orange.*" When they settled on a final image without their desired features, they explained that they would prefer to add them in using Photoshop rather than continue to iterate on the prompt.

I actually found one that actually kind of works.... Other than having no spaceship and being completely blue. ...I'm sure in Photoshop you could add the spaceship... (P7)

6.3. Measuring Satisfaction and Usage

Overall, participants enjoyed using the tool, with 10 out of the 12 of participants responding positively (agree or strongly agree) when asked if they would use DREAMSHEETS again. No

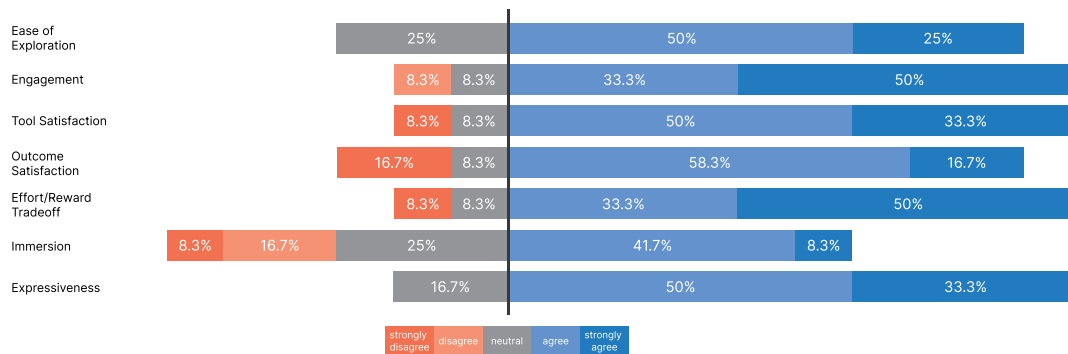


Figure 3.: Distribution of Likert scale responses on creativity support index and satisfaction. See Appendix A for the full questions.

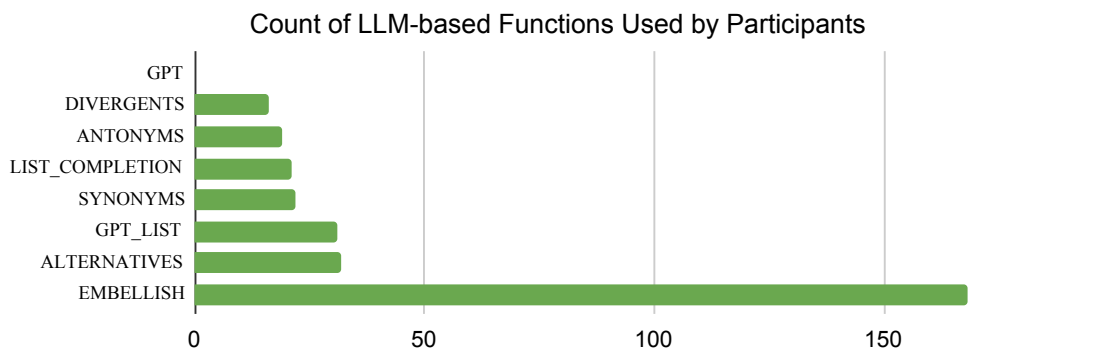


Figure 4.: Number of times participants used each of the LLM-based function during the activity.

participant responded negatively (disagree or strongly disagree) when asked if it was easy to explore with DREAMSHEETS. These results are summarized in Figure 3.

One point of particular dissatisfaction was in missing a sense of “immersion”, the quality of feeling that the tool will melt away, and being able to just *work through the tool*—commonly known as “ready-to-hand”, after Heidegger’s description in *Being and Time* [15]. These results likely reflect the experiences of participants who felt less comfortable working with spreadsheets, especially in attempting more advanced techniques like concatenation and dynamic cell references.

Participants used the LLM-based functions to varying degrees as well, but the EMBELLISH function was most popular by a large margin (approximately 160 uses, compared with fewer 50 for each of the other functions)—see Figure 4 for a full list.

7. Discussion

Overall, our findings suggest that spreadsheets bring a number of useful advantages to design space exploration with TTIs. However, our qualitative analysis suggests that prior experience with either spreadsheets or prompting has a strong mediating impact on how supported participants felt, and how comfortable they were using DREAMSHEETS’s various functions.

7.1. Learning Curves

Spreadsheets are a widely used tool, and likely to be familiar to many users; all of our participants had at least *some* experience with spreadsheets, and most of them reported frequent use.

However, there is a learning curve for comfortably using more advanced spreadsheet features like expanding cells into sequences using auto-fill and working with cell references. Compounded with the task of learning and incorporating novel LLM and TTI functions could make it difficult for less experienced users to truly leverage the benefits of the spreadsheet metaphor towards their creative goals.

Furthermore, even frequent spreadsheet users may stick to using the functions or workflow techniques they are comfortable with, or might tend towards copying example structures given to them, rather than exploring more complex or less familiar workflow options. Some users were explicitly reluctant to embrace the freedom and flexibility afforded by spreadsheets and said they would prefer to be given more structure (P10, P12).

Prior experience in the prompt-engineering and Text-To-Image domain particularly impacted participants’ use of DREAMSHEETS.

DREAMSHEETS did not improve *discoverability* of prompt features nor did it make them explicit. More experienced participants knew what kinds of prompt terms (art movements, camera angles, and artist names) were likely to influence the generated image results in interesting ways; meanwhile, less experienced participants struggled to interpret Stable Diffusion’s “behavior” and what they perceived as the model’s “misunderstanding”.

That said, DREAMSHEETS aided prompt exploration regardless of participant background: it supported participants with a space to *explore* the model’s “knowledge” space, while preserving the history of their explorations, helping them build intuitions and prompt engineering skills.

Even basic spreadsheet features, that is, those simple enough to be accessible to even infrequent spreadsheet users, are valuable in facilitating exploratory research and can make a substantial improvement in supporting creative TTI interaction. These basic affordances include the ability to copy, paste, undo and redo within a structured, but flexibly customizable visual layout that preserves the user’s inputs and the corresponding outputs.

7.2. Integration Trade-Offs

A major advantage of DREAMSHEETS is that it allowed users to pull in and “chain” tools (like LLM functions) to support them throughout creative process and to design their workflow to fit their needs.

P5, who used EMBELLISH to create variations on manually written prompts explicitly identifies this benefit: “What EMBELLISH does... it’s good chaining, right? So, I think that influences your experience.”

While EMBELLISH, the most frequently utilized LLM function, may have perceptively “improved” prompts by introducing more verbose and descriptive language – it did not always make a prompt more effective as a TTI input or as an input to the Stable Diffusion model in particular.

Generating desirable images was our participants’ end-goal in our study, and it’s possible that DREAMSHEETS users would have been better supported if the verbose variations generated by EMBELLISH also tuned outputs to have an effective TTI prompt structure. PromptParrot [43] is an example of a tool that does this.

7.3. Creativity Support

Spreadsheets as an interface provide features that supported our participants’ creative *exploration* particularly well, as well as their *engagement* in the prompt-engineering creative process—qualities

identified in creativity literature as important factors of the creative process that effective creativity support tools should provide for [6, 37].

Even the basic creativity support tool features DREAMSHEETS supports, like preserving history and reversal of actions, represent a significant improvement over the back-and-forth conversation with the “robo-artist” model common to Discord-based TTI services like MidJourney:

Having used these tools before, I like the sheets interface... if you do it in Discord bot - that’s where I used MidJourney before... You don’t know what’s going on, right? On a sheet, it has a bunch of advantages...It’s already recorded... you don’t have to like create all the subfolders, save it...and the storage happens on the sheet itself. ...You can visualize the prompts that you used before. (P5)

...and also...

When I’ve used these tools before... we maybe specify a seed and just hope it works out for the best, but having this selection in front of me where I can just run it all instantly....I think that’s very useful and I can imagine that being used for considering a bunch of different art pieces and then choosing the best one from the selection right in front of you. (P4)

DREAMSHEETS allowed participants to aggregate, in one site, in a visual layout of their choice, any number of concepts as cells: ideas, experiments, useful examples for use as reference, explorations, intermediate and final results, and more. Furthermore, more advanced users are able to construct relationships between cells, construct new cells from the manipulation of existing ones, and explicitly control which values should remain fixed and which should vary across their explorations.

7.4. Lessons for Future Tool Designers

We identify a few major takeaways for designers of future tools.

First, designers of TTI-based systems should consider how supporting functionality (like DREAMSHEETS’ LLM functions) provides appropriate support of prompt design activities. DREAMSHEETS requires users to construct explicit formulas to engage the LLM-based prompt exploration functionality, but doing so was challenging for novice users. Instead, future tools should consider using *example-driven* formula construction, as in Excel’s FlashFill [12] feature: a few prompts (e.g., a list of dog breeds) can serve as an exemplar to drive a program synthesis approach for formula construction (e.g., proposing LIST_COMPLETION over the “dog breed” cells).

Second, designers that seek to create tools that interface with TTI models should consider how they can support the creative exploratory *research* processes that we observed users tending towards. Users approaching prompt-engineering as a creative process want to use experimentation to build their intuition and understanding of the model’s “knowledge” and “vocabulary”—to identify terms to use in their prompts that will have predictable and productive influence on their image generation results.

[This] would have been immensely helpful when we were trying to just like study this thing [Stable Diffusion]... Wow! Yeah, you can see how the guidance affects it going down.

...said P9 as they used cell expansion to generate a columnwise series of images by passing the same prompt, with various “guidance” (*cfg*) values, into the TTI() function.

"That’s fantastic! It’s really fun to see the guidance... to pull this out in seconds is very awesome, to see how the guidance affects the output. (P9)

Building model experience and comprehension through this kind of exploration builds prompt-engineering skills, and is another reason for designers to support user access to past explorations and results.

7.5. Limitations and Future Work

DREAMSHEETS had a number of limitations: no image editing capabilities like inpainting and masking, no choice in image resolution, slow image generation—these all undoubtedly impacted the user experience, and a tool with these features may have yielded different results.

Additionally, our relatively small formative study was not large enough to make statistically significant claims about satisfaction, usability, or the impacts of prior experience on use—though we do explore a few hypotheses in §7 (Discussion) that we encourage future researchers to explore.

We also encourage future designers and researchers to explore alternative ways of constructing 2D matrix-based “small multiples” interfaces for exploring TTI outputs. Further, formula construction, while approachable to experienced spreadsheet users, could be made more straightforward with better autocomplete and a more complete set of examples containing more complex formula constructions.

Finally, we encourage future designers of TTI-based tools to explore explicit methods for supporting user intuition development and an understanding of model capabilities.

8. Conclusion

Using Text-To-Image models to achieve creative goals often involves highly iterative and dynamic exploratory processes that existing systems and tools may fail to effectively support. We present DREAMSHEETS, a creativity support tool that provides a spreadsheet interface as the site for user's interactions with LLMs and TTI, supporting rich exploratory research, history, and the freedom to develop custom workflows. Our user study of 12 participants provides rich insights into how a spreadsheet interface might support or hinder end-users in exploring the TTI generation design space. We found that participants' prior experiences with spreadsheets, LLMs, and TTIs will impact their efficacy as prompt-engineers in DREAMSHEETS, but that even novice users are able to leverage DREAMSHEETS's features towards their engagement in the creative process. We also discuss the implications for the design of future TTI and LLM tools, such as the importance of supporting users in exploratory research processes, allowing them to build prompt-engineering *intuition* and develop their "prompt-language" with a particular model while refining their creative process.

Bibliography

1. M. Agrawala. *Unpredictable Black Boxes are Terrible Interfaces*. 2023. URL: <https://magrawala.substack.com/p/unpredictable-black-boxes-are-terrible> (visited on 04/05/2023).
2. proxima centauri b, surea.i, S. Young, and E. Erratica. *Image Synthesis Style Studies Database by parrotzone.art*. 2023. URL: https://web.archive.org/web/20230325013519/https://docs.google.com/spreadsheets/d/14xTqtuV3BuKDNhLotB_d1aF1BGnDJOY0BRXJ8-86GpA/edit#gid=0 (visited on 04/04/2023).
3. M. J. Bitner, A. L. Ostrom, and F. N. Morgan. "Service blueprinting: a practical technique for service innovation". In: *California management review* 50:3, 2008, pp. 66–94.
4. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].
5. M. BURNETT, J. ATWOOD, R. WALPOLE DJANG, J. REICHWEIN, H. GOTTFRIED, and S. YANG. "Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm". In: *Journal of Functional Programming* 11:2, 2001, pp. 155–206. DOI: 10.1017/S095679680003828.
6. E. Cherry and C. Latulipe. "Quantifying the Creativity Support of Digital Tools through the Creativity Support Index". In: *ACM Trans. Comput.-Hum. Interact.* 21:4, 2014. ISSN: 1073-0516. DOI: 10.1145/2617588. URL: <https://doi.org/10.1145/2617588>.
7. E. H.-h. Chi, J. Konstan, P. Barry, and J. Riedl. "A Spreadsheet Approach to Information Visualization". In: *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*. UIST '97. Association for Computing Machinery, Banff, Alberta, Canada, 1997, pp. 79–80. ISBN: 0897918819. DOI: 10.1145/263407.263513. URL: <https://doi.org/10.1145/263407.263513>.
8. J. J. Y. Chung, S. He, and E. Adar. "The Intersection of Users, Roles, Interactions, and Technologies in Creativity Support Tools". In: *Designing Interactive Systems Conference 2021*. DIS '21. Association for Computing Machinery, Virtual Event, USA, 2021, pp. 1817–1833. ISBN: 9781450384766. DOI: 10.1145/3461778.3462050. URL: <https://doi.org/10.1145/3461778.3462050>.

9. J. Frich, L. MacDonald Vermeulen, C. Remy, M. M. Biskjaer, and P. Dalsgaard. “Mapping the Landscape of Creativity Support Tools in HCI”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Association for Computing Machinery, Glasgow, Scotland Uk, 2019, pp. 1–18. ISBN: 9781450359702. DOI: 10.1145/3290605.3300619. URL: <https://doi-org.libproxy.berkeley.edu/10.1145/3290605.3300619>.
10. P. B. Gerard Serra Oriol Domingo. “A programmable interface for creative exploration”. In: *Machine Learning for Creativity and Design Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022. URL: https://neuripscreativityworkshop.github.io/2022/papers/ml4cd2022_paper19.pdf.
11. E. L. Glassman, T. Zhang, B. Hartmann, and M. Kim. “Visualizing API Usage Examples at Scale”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2018, pp. 1–12. ISBN: 9781450356206. URL: <https://doi.org/10.1145/3173574.3174154>.
12. S. Gulwani. “Automating string processing in spreadsheets using input-output examples”. In: *ACM Sigplan Notices* 46:1, 2011, pp. 317–330.
13. B. Hartmann, L. Yu, A. Allison, Y. Yang, and S. R. Klemmer. “Design as Exploration: Creating Interface Alternatives through Parallel Authoring and Runtime Tuning”. In: *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. UIST ’08. Association for Computing Machinery, Monterey, CA, USA, 2008, pp. 91–100. ISBN: 9781595939753. DOI: 10.1145/1449715.1449732. URL: <https://doi.org/10.1145/1449715.1449732>.
14. J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. “Graphical histories for visualization: Supporting analysis, communication, and evaluation”. In: *IEEE transactions on visualization and computer graphics* 14:6, 2008, pp. 1189–1196.
15. M. Heidegger, J. Macquarrie, and E. Robinson. “Being and time”. In: 1962.
16. C. Kulkarni, S. Druga, M. Chang, A. Fiannaca, C. Cai, and M. Terry. “A Word is Worth a Thousand Pictures: Prompts as AI Design Material”. In: *arXiv preprint arXiv:2303.12647*, 2023.
17. B. Lee, S. Srivastava, R. Kumar, R. Brafman, and S. R. Klemmer. “Designing with interactive example galleries”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 2257–2266.
18. M. Levoy. “Spreadsheets for images”. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 1994, pp. 139–146.
19. P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. 2021. arXiv: 2107.13586 [cs.CL].

20. V. Liu and L. B. Chilton. “Design Guidelines for Prompt Engineering Text-to-Image Generative Models”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI ’22. Association for Computing Machinery, New Orleans, LA, USA, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3501825. URL: <https://doi.org/10.1145/3491102.3501825>.
21. V. Liu, J. Vermeulen, G. Fitzmaurice, and J. Matejka. *3DALL-E: Integrating Text-to-Image AI in 3D Design Workflows*. 2022. arXiv: 2210.11603 [cs.HC].
22. A. Lunzer and K. Hornbæk. “Subjunctive Interfaces: Extending Applications to Support Parallel Setup, Viewing and Control of Alternative Scenarios”. In: *ACM Trans. Comput.-Hum. Interact.* 14:4, 2008. ISSN: 1073-0516. DOI: 10.1145/1314683.1314685. URL: <https://doi.org/10.1145/1314683.1314685>.
23. J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. “Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation”. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’97. ACM Press/Addison-Wesley Publishing Co., USA, 1997, pp. 389–400. ISBN: 0897918967. DOI: 10.1145/258734.258887. URL: <https://doi.org/10.1145/258734.258887>.
24. J. Matejka, M. Glueck, E. Bradner, A. Hashemi, T. Grossman, and G. Fitzmaurice. “Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2018, pp. 1–12. ISBN: 9781450356206. URL: <https://doi.org/10.1145/3173574.3173943>.
25. B. Moggridge. *Designing Interactions*. The MIT Press, 2006. ISBN: 0262134748.
26. A. Mohiuddin and R. Woodbury. “Interactive Parallel Coordinates for Parametric Design Space Exploration”. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA ’20. Association for Computing Machinery, Honolulu, HI, USA, 2020, pp. 1–9. ISBN: 9781450368193. DOI: 10.1145/3334480.3383101. URL: <https://doi.org/10.1145/3334480.3383101>.
27. D. Nagy, L. Villaggi, D. Zhao, and D. Benjamin. “Beyond heuristics: a novel design space model for generative space planning in architecture”. In: 2017.
28. G. Parsons. *The DALL-E 2 Prompt Book*. 2022. URL: <https://dallery.gallery/the-dalle-2-prompt-book/> (visited on 09/15/2022).
29. A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. arXiv: 2204.06125 [cs.CV].
30. E. Rawn, J. Li, E. Paulos, and S. Chasins. “Understanding Version Control as Material Interaction with Quickpose”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2023.

31. M. Resnick, B. Myers, K. Nakakoji, B. Shneiderman, R. Pausch, T. Selker, and M. Eisenberg. “Design Principles for Tools to Support Creative Thinking”. In: 2005. DOI: 10.1184/R1/6621917.v1. URL: https://kilthub.cmu.edu/articles/journal_contribution/Design_Principles_for_Tools_to_Support_Creative_Thinking/6621917.
32. D. Ritchie, A. A. Kejriwal, and S. R. Klemmer. “d. tour: Style-based exploration of design example galleries”. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 2011, pp. 165–174.
33. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10684–10695.
34. A. Salerno. *4.2 Gigabytes, or: How to Draw Anything*. 2022. URL: <https://andys.page/posts/how-to-draw/> (visited on 04/04/2023).
35. V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. Bers, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. *Multitask Prompted Training Enables Zero-Shot Task Generalization*. 2021. DOI: 10.48550/ARXIV.2110.08207. URL: <https://arxiv.org/abs/2110.08207>.
36. A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik. “Interactive Design Space Exploration and Optimization for CAD Models”. In: *ACM Trans. Graph.* 36:4, 2017. ISSN: 0730-0301. DOI: 10.1145/3072959.3073688. URL: <https://doi.org/10.1145/3072959.3073688>.
37. B. Shneiderman. “Creativity Support Tools”. In: *Commun. ACM* 45:10, 2002, pp. 116–120. ISSN: 0001-0782. DOI: 10.1145/570907.570945. URL: <https://doi.org/10.1145/570907.570945>.
38. J. O. Talton, D. Gibson, L. Yang, P. Hanrahan, and V. Koltun. “Exploratory Modeling with Collaborative Design Spaces”. In: *ACM SIGGRAPH Asia 2009 Papers*. SIGGRAPH Asia ’09. Association for Computing Machinery, Yokohama, Japan, 2009. ISBN: 9781605588582. DOI: 10.1145/1661412.1618513. URL: <https://doi.org/10.1145/1661412.1618513>.
39. E. R. Tufte. *Envisioning information*. Vol. 199. o. Graphics Press Cheshire, CT, 1993.
40. R. F. Woodbury and A. L. Burrow. “A typology of design space explorers”. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 20:2, 2006, pp. 143–153. DOI: 10.1017/S0890060406060136.
41. T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai. “PromptChainer: Chaining Large Language Model Prompts through Visual Programming”. In: *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022.

42. T. Wu, M. Terry, and C. J. Cai. “AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts”. In: *Proceedings of the 2022 CHI conference on human factors in computing systems*. 2022.
43. S. Young. *COG Prompt Parrot*. 2022. URL: <https://github.com/kyrick/cog-prompt-parrot>.
44. L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, N. Shireen, and M. Terry. “GEM-NI: A System for Creating and Managing Alternatives In Generative Design”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Association for Computing Machinery, Seoul, Republic of Korea, 2015, pp. 1201–1210. ISBN: 9781450331456. DOI: 10.1145/2702123.2702398. URL: <https://doi.org/10.1145/2702123.2702398>.
45. J. Zamfirescu-Pereira, R. Wong, B. Hartmann, and Q. Yang. “Why Johnny can’t prompt: how non-AI experts try (and fail) to design LLM prompts”. In: *Proceedings of the 2023 CHI conference on human factors in computing systems (CHI’23)*. 2023.

Appendices

A. Likert Questions

Area of Focus	Full Question
Ease of Exploration	It was easy for me to explore many different options, ideas, designs, or outcomes without a lot of tedious, repetitive interactions
Engagement	I was very absorbed/engaged in this activity - I enjoyed it and would do it again
Tool Satisfaction	I could see myself using this tool again
Outcome Satisfaction	I was satisfied with the end results I have chosen
Effort/Reward Tradeoff	What I was able to produce was worth the effort required to produce it
Immersion	While I was doing the activity, the tool “disappeared” and I was able to concentrate on the activity
Expressiveness	I was able to be very expressive and creative while doing the activity

The full questions asked to participants after completing the user study.

B. Tutorial Screenshots

B.1. TTI Introduction Sheet

	A	B	C	D	E	F	G	H	I
1	TTI Introduction Sheet	<code>TTI(prompt, seed=GLOBAL_SEED, guidance=13)</code>	guidance: [0, 35]						
2	For this study, we ask you to generate the prompt outside the TTI function call. This will help see what prompt is when using spreadsheet formulas to prompt engineer, as there is no way to see the full prompt when formulating inside the TTI function call.	Text-to-image (TTI) function to generate an image using Stable Diffusion based the given prompt. Function returns a URL, so remember to wrap it with an IMAGE function to view the image.	How strictly the diffusion process adheres to the prompt text (higher values keep your image closer to your prompt)How strictly the diffusion process adheres to the prompt text (higher values keep your image closer to your prompt)						
3	running		guidance/seed		1111	2222	3333	4444	5555
4	dog running on a hill			1					
5			5						
6			9						
7			13						
8			17						
9			21						

A screenshot of the TTI introduction sheet for our user study.

B.2. GPT Functions' Introduction Sheet

A1	A	B	C	D	E	F	G	H	I	J	K	L
1	GPT(prompt)	GPT_LIST(phrase, length=5)	LIST_COMPLETION(list, length=5)	SYNONYMS(word, length=5)	ANTONYMS(word, length=5)	DIVERGENTS(word, length=5)	ALTERNATIVES(word/phrase, length=5)	EMBELLISH(phrase)				
2	GPT_LIST is a structured GPT-3 call to make sure what it gives back is a list of items based on the phrase given. The resulting elements will populate the calls below the function call; therefore, make sure there are nothing already in those calls.	GPT_LIST is a structured GPT-3 call to make sure what it gives back is a list of items based on the phrase given. The resulting elements will populate the calls below the function call; therefore, make sure there are nothing already in those calls.	LIST_COMPLETION takes in a list of items and returns a list that contains elements that are similar to the given list.	SYNONYMS takes in a word and returns synonyms of the given word.	ANTONYMS takes in a word and returns antonyms of the given word.	DIVERGENTS takes in a word and returns words that are different but related words.	ALTERNATIVES takes in a word or phrase and returns alternative ways to say the given word or phrase.	EMBELLISH takes in a phrase and will return an embellished version of the given phrase.				
3	GPT is a function that uses GPT-3 to help with prompt engineering and forms the basis of the other specific GPT-related functions.	GPT_LIST is a structured GPT-3 call to make sure what it gives back is a list of items based on the phrase given. The resulting elements will populate the calls below the function call; therefore, make sure there are nothing already in those calls.	LIST_COMPLETION takes in a list of items and returns a list that contains elements that are similar to the given list.	SYNONYMS takes in a word and returns synonyms of the given word.	ANTONYMS takes in a word and returns antonyms of the given word.	DIVERGENTS takes in a word and returns words that are different but related words.	ALTERNATIVES takes in a word or phrase and returns alternative ways to say the given word or phrase.	EMBELLISH takes in a phrase and will return an embellished version of the given phrase.				
4		types of animals	happy	happy	mountain							
5		=GPT_LIST(B3, 3)	=LIST_COMPLETION(B5:B7)	=SYNONYMS(D3)	=ANTONYMS(E3)	=DIVERGENTS(F3)	=ALTERNATIVES(G3)	=EMBELLISH(H3)				
6		lion	cheetah	pleased	sad	valley	Canine sprinting over a hill	The dog was running joyfully up the hill, its tail wagging in the wind.				
7		tiger	jaguar	joyful	unhappy	ocean	Pooch bounding up a slope					
8		leopard	panther	content	miserable	desert	Dog dashing across a peak					
9			lynx	delighted	depressed	tundra	Mutt rising up a mountain					
10			cougar	gleeful	gloomy	jungle	Hound scampering on a rise					
11		mountain	valley	ocean	desert	forest						
12	dogs	dogs on a mountain	dogs on a valley	dogs on a ocean	dogs on a desert	dogs on a forest						
13	cats	cats on a mountain	cats on a valley	cats on a ocean	cats on a desert	cats on a forest						
14	frogs	frogs on a mountain	frogs on a valley	frogs on a ocean	frogs on a desert	frogs on a forest						
15	horses	horses on a mountain	horses on a valley	horses on a ocean	horses on a desert	horses on a forest						
16	deer	deer on a mountain	deer on a valley	deer on a ocean	deer on a desert	deer on a forest						

A screenshot of the GPT functions' introduction sheet for our user study.