Noise Pilot: Enabling Artistic Workflow Composition with Diffusion-Based Image Generation

James Smith
University of California, Berkeley
Berkeley, California, USA
james.smith@berkeley.edu

Shm Garanganao Almeda
University of California, Berkeley
Berkeley, California, USA
shm.almeda@berkeley.edu

Timothy J. Aveni
University of California, Berkeley
Berkeley, California, USA
tja@berkeley.edu

Anya Agarwal
University of California, Berkeley
Berkeley, California, USA
anya.agarwal@berkeley.edu

Bjoern Hartmann
University of California, Berkeley
Berkeley, California, USA
bjoern@eecs.berkeley.edu

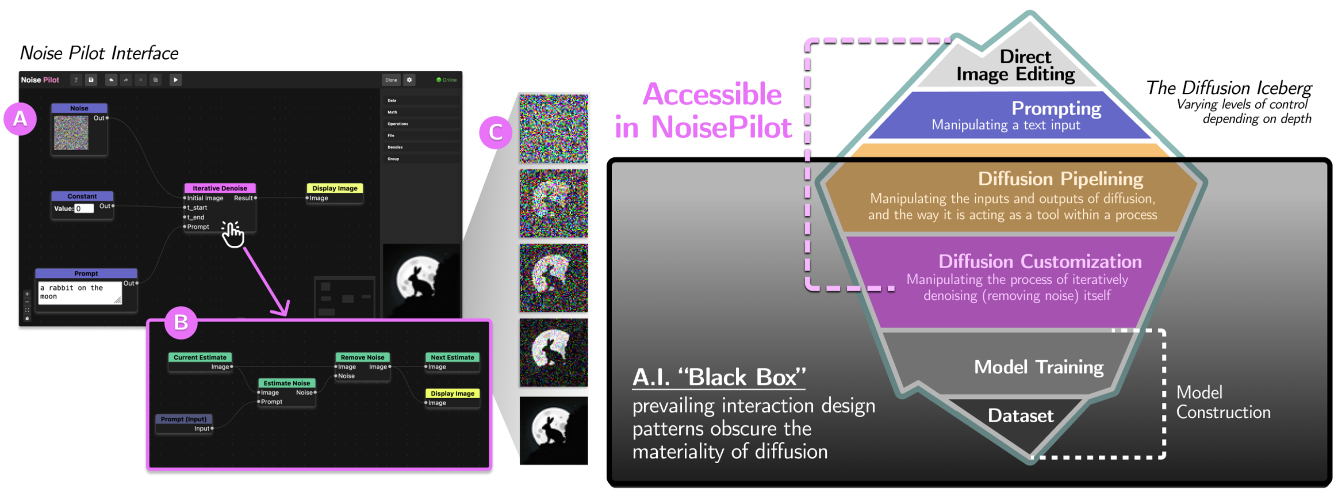

Figure 1: Left: Noise Pilot interface. (A) View of the base image generation file. The user can modify how diffusion happens by double clicking the Iterative Denoise node. (B) Within the node there is another node canvas that controls how diffusion occurs for this image. (C) Example outputs in the UI of generating an image. Right: Representation of the different instruments for manipulating image diffusion. Noise Pilot allows artists to take creative actions at a deeper level than other tools.

Abstract

Creativity support tools (CSTs) increasingly include image-generation features. The underlying diffusion models enact a particular image diffusing process that AI CSTs tend to obscure within a black-box. Artists' creative control is limited to indirect manipulation (prompting), chaining these "black-boxes" together, or using ML-engineering skills to build custom black-boxes. Seeking to maintain the low-threshold offered by prompting, while raising the ceiling of expressive interactions, we built Noise Pilot: a multi-layered approach to supporting diffusion-based creative processes at three levels of depth. We used Noise Pilot as a probe to study the artistic processes of 9 artists over a 2-week period. Artists engaged with diffusion at different levels of manipulative depth and crafted reusable artifacts to enact bespoke diffusion processes; some produced results impossible to achieve with prompting alone. We discuss how black-box AIs in CSTs limit creative power, and propose subverting this by favoring visibility over obscurity, and materiality over personification.

CCS Concepts

• **Human-centered computing** → Graphical user interfaces; **Empirical studies in HCI**.

Keywords

Generative AI, Interaction Design, Node Based Programming

ACM Reference Format:
James Smith, Shm Garanganao Almeda, Timothy J. Aveni, Anya Agarwal, and Bjoern Hartmann. 2026. Noise Pilot: Enabling Artistic Workflow Composition with Diffusion-Based Image Generation. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3772318.3790531

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3790531

1 Introduction

As new algorithmic techniques in image technology emerge, the design of new interfaces and interactions follows, enabling new modes of artistic expression. As these new opportunities and interfaces transform artistic workflows, HCI researchers are called to revisit the way we conceptualize and support creative process.

One such paradigm-shifting advancement is the emergence of text-conditioned image-generation models. The multi-modal affordances of these models — from supporting text-prompt guided image generation [27], to visual style transfer [20] and beyond — have significant implications for the future of creative image manipulation. Text-to-image **diffusion models**, which produce images by iteratively removing noise as guided by a text prompt have been particularly influential — users can simply describe an image that they want and the model will generate it for them. This type of interaction affords low thresholds for getting started as most people are able to describe an image they want and see an immediate result. The power of these capabilities has led to widespread integration of prompt-centered image-generation into interfaces like ChatGPT [50] and Photoshop's generative fill feature.

Effectively prompting these "black boxes" has its challenges — empirical studies into the difficulties associated with prompting [10, 53, 71] have revealed that people lack specific vocabulary for effectively representing their ideas as prompts. One option for designing around these limitations is by effective prompting *explorations*, to support iteratively probing the model inside these black boxes through prompts — the approach of recent HCI literature DreamSheets [2] and Promptify [7].

One successful approach to supporting a fine level control over computational creative processes is through node-based programming interfaces. These are frequently used by people who have a mix of creative and computational backgrounds (e.g., creative technologists and technical artists). We'll use "computational artists" (CAs) to denote this user group. There are many examples of such interfaces, ranging from a variety of domains such as architecture [5], 3D graphics [17], visual effects [60], audio [1], as well as prior literature in HCI exploring new artist-oriented programming interfaces [33]. A popular recent tool for diffusion, ComfyUI [54], also utilizes this paradigm to allow users to programmatically specify inputs, load different machine learning models, and pipeline outputs of one model into another. However, generating images with diffusion models is also treated as a black box within this interface, leading users to encounter some of the same struggles as with prompting tools.

It turns out diffusion models are capable of generating images that are impossible to prompt for. There are a number of techniques from the machine learning literature [21, 22] showing how customizing the internals of the black box allows for creating entirely new types of images. Customizing the way that these techniques generate images is currently only accessible to those with software engineering skills, requiring the user of either pre-made custom black-box tools, or the ability to write Python code.

We ask the question: Can a multi-layered [59] interface design that simultaneously offers three different levels of interacting with diffusion (See Figures 1 and 2) allow computational artists to successfully expand their useful artistic control and exploration of diffusion models?

To explore this question, we designed, implemented and studied Noise Pilot, a node-based programming environment for authoring diffusion-based image generation workflows. A key innovation is that Noise Pilot implements the standard diffusion process as a series of nodes in the tool itself, making it directly editable by users. This representation allows artists to customize the diffusion process and flexibly compose it with other image operations into bespoke, reusable image generation workflows.

We demonstrate the expressive power of this environment by reimplementing 4 research papers on novel diffusion techniques (visual anagrams [22], factorized diffusion [21], image-to-image translation [32], generative fill [42]), and demonstrate an additional technique not found in the research literature (controlled tiled generation).

We then study how this tool might impact creative practice through a two-week deployment with 9 artists. We analyze artists' strategies using Noise Pilot to support art-making, finding that they successfully used and moved across all three levels of manipulative depth. We report on how artists were empowered by increased transparency and control over the diffusion process and discuss how this kind of support can help demystify and empower artists' interactions with black-box systems.

We apply our findings to a discussion of the implications for the future development of AI-based creativity-supportive systems; we discuss the crucial need to shift *away* from disempowering artists with interface designs that supplant their material agency, and *towards* empowering artists with the option to engage with diffusion as a kind of digital material — one they can manipulate, combine, or reject, as they might any other art-making medium. The contributions of this paper are as follows:

- Noise Pilot, a multi-layered creativity support tool for diffusion-based image generation.
- Findings from a 2-week extended study, which we connect to literature and future work.
- A discussion on the implications of designing better tools to empower artists by supporting material interactions with diffusion.

2 Background

Understanding the specifics of how a diffusion model generates images is important background information for this work. There are a variety of ways in which machine learning models can generate images. Noise Pilot uses models trained using the Denoising Diffusion Probabilistic Models (DDPM) [27] approach, which is currently the most common method for training diffusion models. Reference Figure 3 for a graphical representation of this process (based on a figure from the original DDPM paper).

Generating images using DDPM-based models is an iterative process that takes T steps. The diffusion model is represented as a function p_θ (known as the reverse process) that predicts the noise in a noisy image x_t. The predicted noise is removed from x_t to produce x_{t-1}, which is a slightly less noisy image (See Figure 3).

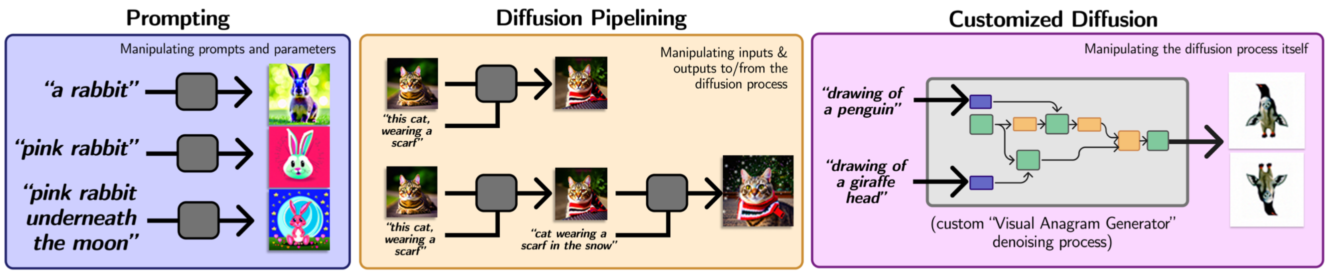

Figure 2: Examples of diffusion manipulation processes supported by Noise Pilot. (1) With prompting, users make changes to the text prompt and other input parameters to manipulate the image outputs, as supported by many tools. (2) With diffusion pipelining, users compose workflows that manipulate processes before and after a diffusion process, while still treating diffusion models as a black box. (3) With diffusion customization, users directly modify the diffusion algorithm itself, allowing granular control over *how* the image is generated.

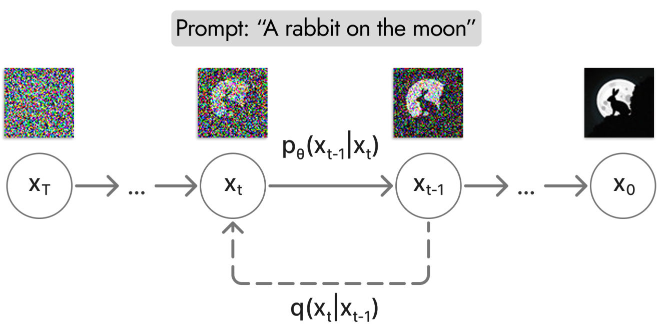

Figure 3: Diffusion trained with DDPM estimates noise in an image at time t as a function p_θ, conditioned on a prompt, which is used to produce the next slightly-less-noisy image at x_{t-1}. This process repeats until all of the noise is removed and we are left with an image that represents the prompt. This figure is our recreation of the original explanation of the denoising loop in the DDPM paper [27], Figure 2.

Throughout the paper we will refer to this iterative process as **iterative denoise** or a **denoise loop**.

In the general base case of prompting for an image, the process starts at $t = T$ with an image of pure noise, and performs T denoise operations until $t = 0$. Timesteps are represented in this order due to the fact that these models are trained in reverse (starting with noiseless images, and iteratively adding noise known as a "noise schedule" which is defined by the function q). Not all methods for generating images using diffusion use the $[T, 0]$ bounds, however, as we discuss later in our survey of image generation techniques that utilize DDPM (see Section 3).

3 Related Work

Our research builds on and extends prior work on interfaces for AI image generation in general, and on algorithms for customizing diffusion-based image generation in particular. We also connect to the broader research area of creative programming environments, and to research on the impact of GenAI on creative and artistic processes. We discuss each area in turn.

3.1 AI Image Generation

3.1.1 User Interfaces for image generation. The powerful creative potential of diffusion models themselves is evident — but limited by the space of interfaces designed to allow users control over them. Mainstream commercial AI tools such as DALL-E [51], Midjourney [45], and ChatGPT [50] provide access to these models through a simple prompt interface. Interfaces for image-to-image translation, like Draw Things [66] or NVIDIA Canvas [47], allow the user to provide a starting image in addition to a prompt input.

While a recurring interface design theme is to present *prompting* as the primary interaction method for controlling these models, writing an effective prompt is not straightforward [71] and can require skill, technique, and iteration. Empirical studies on creative use of text-to-image models [10, 35, 52, 53, 55] have recognized the potential of prompts as mediums for art-making but they also identify the limitations of prompting as a creative interaction technique and find that practitioners, especially those who are less interested in the creative "constraints" of prompting, often desire more control over text-to-image models.

We are beginning to see a wave of interfaces that attempt to leverage AI models for supporting creative processes. This includes interfaces that improve text-to-image model design space exploration [2, 7, 13]. Other popular approaches range from utilizing the multi-modal nature of models to refine outputs [56, 70], to helping users extract relevant keywords to construct better prompts [12, 40, 65], to using the ability for models to generate image variations as part of the exploration process [25, 61].

Beyond prompting based interfaces, there also exist interfaces focused on parameter tuning, and node-based UIs. Parameter-tuning interfaces afford more control by exposing model parameters as UI (e.g., AUTOMATIC1111 [6]). Node-based tools like ComfyUI allow users to assemble sequences of nodes into workflows, to compose custom ways of generating images. The interface language of ComfyUI mirrors the domain expertise of machine learning and is grounded in data operations: nodes are parameterized by conditionings, models, latents, VAEs, and CLIPs. Custom diffusion techniques can be implemented as community-developed custom nodes in ComfyUI. However, creating or modifying custom nodes requires much deeper programming expertise than using existing nodes.

We believe ComfyUI is a great step in the direction of allowing for integrating diffusion models into artistic workflows. However, there is currently no ability in the tool to customize the way that diffusion happens, necessitating a custom tool for our study. We also made the choice to abstract away some of the complexities present in ComfyUI to avoid requiring domain knowledge, focusing purely on simple data types (prompts and images) and a set of default nodes that more closely resemble creativity applications.

3.1.2 Techniques for customizing diffusion. There are many examples from the computer vision literature that show how modifying the diffusion process can lead to interesting and novel images. These techniques inspire the creation of this work, but with the goal of making these modifications accessible to computational artists. Below we describe the most relevant examples.

A common technique supported by a number of diffusion based tools is Image-to-Image Translation [32]. This technique uses an input image as a starting point and fills in details based on a prompt. This technique can support a variety of input images ranging from photographs to crude sketches. Generative fill techniques such as Inpainting [42] allow for generating parts of images by diffusing only within a mask, while trying to blend with the rest of the image at the mask boundary. This allows for diffusion-based image alterations.

Visual Anagrams [22] from Geng et al. contributed a technique for generating optical illusions with diffusion models. Their insight is that by prompting the model for two noise estimates, where each noise estimate can encode a different constraint on the final output image, they can consistently generate images that cannot be achieved through prompting alone. Factorized Diffusion [21], also by Geng et al., uses a similar approach to generate hybrid images [49], which are images that have two meanings depending on how they are viewed.

There are a number of other approaches that afford control over diffusion model outputs. Techniques such as ControlNet [72] attach a machine learning model to the input portion of a diffusion model, allowing users to use other forms of media to steer image diffusion. InstructPix2Pix [9] takes a similar approach but allows for the editing of images using human instruction. In terms of controlling the output, techniques like Low-Rank Adaptation (LoRA) [29] allow for efficient re-training of model outputs, allowing users to create custom models to produce images in specific styles. All of these approaches either require knowledge of how to train these models or access to a pre-trained model trained in the media that they desire, and therefore remain outside the scope of what our tools seeks to address.

3.2 Creative Programming Environments

Software code can be a medium for creative and exploratory practice, whether that code is fully generated (as in visualization widgets in DynaVis [68]), generated with user-specified interconnections (as in Spellburst [4]), or written fully manually (as in many real-world creative tools, such as Grasshopper [5], shader graph editors [17], Houdini [60] for visual effects, and MaxMSP [1] for audio). Many of these tools use visual and node-based programming environments to situate program information. In this paradigm, users control the authorship of code which then generates artifacts, with users directing behavior at a high level for this generation rather than operating directly on the artifact itself.

HCI systems research often explores these creative programming environments through qualitative studies of diverse digital mediums, such as dynamic paintbrushes [37], 3D animations [43], and design prototyping [26]. Li et al. [38] discuss a balance between the efficiency afforded by software assistance and the rigidity imposed by automation; artists' own customization of their software tools may help them better choose the boundaries of their creative practice. Programming is also a means of abstraction; controlling these abstractions can empower artists to move between different levels of engagement with their creative work [39].

3.3 Creativity Support Systems and AI's Emerging Role

A significant body of HCI research is dedicated to researching how to design tools to better support creativity, and their impact on creative process [18, 19]. Shneiderman, in his 2007 work on creativity support tools (CSTs), identifies a number of design principles, such as *support exploratory search* and *design with low thresholds, high ceilings, and wide walls* [59]. He proposes using multi-layered interface designs to help with these problems, where the first layer is accessible to novices but also affords the ability to move up to higher layers as users become more experienced in the tool. More recent CST research has tackled broader sociocultural and cognitive phenomena shaping creative practice. Li et al. highlight how designing tools that make aspects of a creative process "easy" can shift power *away* from artists, by automating or abstracting away from elements of their process that they are artistically interested in [39]. They propose empowering artists the flexibility to "compose tools in an efficient workflow, or refuse tools by replacing them with different ones".

One way that researchers have proposed empowering artists in their experiences making with digital tools is by drawing on the values and experiences of art-making with physical materials. For example, Chung & Adar's *PromptPaint* allows users to prompt using "paint-medium-like" interactions; this allowed users to "iteratively shape" their creations and feel a sense of creative "ownership" over the output [13]. This notion of "shaping" materials is valued in craft and art-making [57], yet remains under-supported in diffusion systems that support prompting alone, centering indirect manipulation of image diffusion through text, rather than direct engagement with the medium's materiality.

Introducing AI-powered tooling into creative practices can affect artists' agency over their processes, especially when AI tools are affecting creative artifacts directly. Generative AI has enabled recent research [48, 63] to explore direct, *in situ* modifications of artifacts by AI tools, finding that users often prefer to manage the balance of their control over their work. These works also find that AI suggestions may serve to inject unpredictability into creative process or act as an "icebreaker" to justify divergent ideation. Anderson et al. [3] demonstrate that offloading creative ideation effort to AI tools may result in decreased originality relative to other users, but also that these tools may provoke users to explore a larger diversity of ideas within their own work. Other approaches inject agents that interact with the software's user(s), rather than directly with a creative artifact, by providing critique [15, 16] or offering suggestions that need to be incorporated by users [23]. This approach can similarly encourage divergent ideation at the risk of compromising authors' individual ownership.

AI-based art generation tools inherently contest users' ownership over their artistic creations, since models will introduce artifacts from their pre-training. This constraint, though it may limit artists' agency in some forms, suggests a creative opportunity: users may *find* existing patterns that are worth bringing into an art piece through collage or synthesis. As prior works demonstrates in text-to-image prompting [10], storytelling [64], and poetry [36], the act of discovery [46] can itself be a core creative practice when working with generative AI models. Almeda et al. [2] identify that practitioners can also conduct highly structured explorations of the space of AI output behaviors with the help of computational tools.

Offering *explanations* for AI behavior may provide insight into the nature of AI decision-making assistance and reduce the risk of overreliance on AI outputs [69]. AI creativity tools that are too opaque may steer users away from reflective exploration, pushing instead toward a less flexible, more goal-oriented approach.

By unpacking the diffusion process, Noise Pilot offers a means of explanation to users, showing the path that an algorithm takes through the diffusion space both before and after the user has made changes to the diffusion program. In this design, the user sees the process of the AI system approaching the final result through denoising, offering a path for users to consider their process in concert with a more granular view of how the AI model affects the output.

4 Design Goals

The goal of this work is to see what, and how, artists create when offered more control over the diffusion process. We approach the creation of this tool with the following design principles informed by our understanding of the creative process and findings from prior work.

4.1 Enabling Reflective Conversation Between Artist and Material

It is unlikely that users of our system will have experience modifying the iterative denoise process. Therefore we should embed within it the ability for artists to build an intuition about how taking actions in the tool will effect image outputs. This is similar to Schön's "knowing-in-action" [58], the tacit knowledge available to a practitioner based on their experience. The creative process is a reflective conversation between a designer and their material; digital materials must be designed to have qualities that support flexible creative exploration [62].

To enable this, we utilize the principles of direct manipulation interfaces to support incremental and reversible actions [59] to support what Schön calls "experiments". These are actions taken to learn about the limitations of a material. We believe node-based programming interfaces provide a reasonable workspace where artists could make changes to the computation, observe the effects, and reason about how the change produced the observed outputs.

4.2 Support Glass Box Abstractions

We've previously discussed the potential issues with integrating black-box systems as part of the creative process. We take inspiration from Tom Igoe's description of well-designed code libraries as "glass-box enclosures" [30], which encapsulate the internals of how they operate and provide a convenient-to-use interface, but also afford the ability to open them up, understand them, and change them. We propose that AI systems integrated into creative practice should be designed as "glass-box" abstractions to support the same goals.

Our aim is to give artists control over the iterative denoise process, the DDPM algorithm, which is a temporal process that executes a program at each step based on a noise schedule. To effectively do this, we will need to provide visibility into how the process executes so artists can observe how their changes affect the output at each step of the process. By making this observation, they can close the loop on their "experiments" and build an intuition of how their changes affect the output.

4.3 Balance Familiar and Novel Conceptual Models

Leveraging existing conceptual models about how software operates provides an easy on-ramp for users. However, more expressive power in interfaces often comes from deviating from familiar conceptual models and introducing new models that have to be learned and understood. Introducing new functionality is thus a balancing act between familiarity and novelty.

Many computational visual artists are familiar with both image processing operations, and node-based tools. However, few will be familiar with the internals of diffusion loops. We therefore adopt the overall interface language of node-based visual programming with image processing operations as building blocks, while injecting novel nodes that operationalize the diffusion loop functionality within that same conceptual framework.

5 Noise Pilot

Noise Pilot is a node-based image generation tool for composing image manipulation operations along with the noise estimation outputs from diffusion models. The interface allows users to place and connect nodes, which represent operations to manipulate images before, during, and after a diffusion process. See Figure 1, Left for a screenshot of the interface. The node-based paradigm was chosen to support our design goal of materiality, allowing artists to try using new nodes in an exploratory way while allowing actions to be easily reversible.

We represent the DDPM process (which we refer to as Iterative Diffusion or a Diffusion Loop) as nodes in the interface, which presents diffusion as equal and composable with other image processing operations, as opposed to of a complete end-to-end process from prompt to image. This form of interface allows users to create

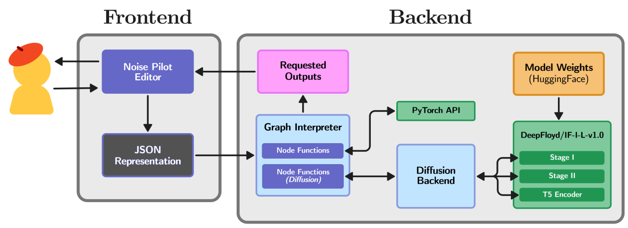

Figure 4: System design of Noise Pilot. Blue represents software modules, pink represents data that is either generated or loaded, and green represents external libraries. The user interacts with a graph editor where they can author an image generating process. These graphs are compiled into a JSON representation which is sent to the server, where it is interpreted by a module that calls the diffusion backend.

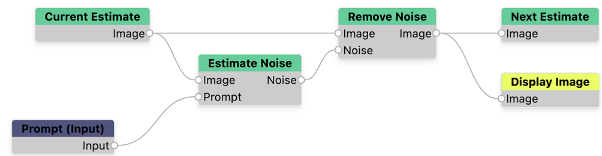

Figure 5: Screenshot from Noise Pilot's implementation of default iterative denoise. This is a depiction of the process shown in Figure 3. This loop is run N times based on parameters specified in the interface.

images that go beyond prompting, because users can compose custom pipelines or customize the denoise loop itself. This is counter to a design that presents prompting diffusion models as a central or primary interaction that might replace or supplant the need for other image manipulation processes entirely.

5.1 Features

The interface for Noise Pilot consists of four parts (See Figure 1, Left). Files in Noise Pilot are referred to as *workspaces*, and are represented by an infinite canvas where the user can construct a graph of nodes and edges. Nodes have parameters that can accept data (inputs), produce data (outputs), or both. The input parameters and return value of the node are each rendered as circular targets along the side of each node. Edges can be created by dragging the mouse between them, connecting an input to an output.

Nodes are added to the canvas by dragging them from a side panel, which displays all of the node types in categories (see Section 5.2 for a description of these categories).

A toolbar gives users access to commonly used features (also available via keyboard shortcut) such as undo, redo, deleting nodes, and duplicating nodes. Clicking the play button in the toolbar will *run* the workspace, executing all of the nodes in the graph.

Images can be displayed in an image preview panel via a special *Display Image* node. This gives users the option to choose whether they would like to visualize each step of a diffusion procedure in progress, supporting our design goal of glass-box abstractions, or to only display final image outputs by simply removing the display node from the denoise loop. This allows users to customize the behavior of the interface to their use case. Displayed images are saved to a gallery, where they can be viewed and downloaded later in JPEG format or (if the user selects a series of images) as an animated GIF.

5.2 Node Types

We implement a variety of nodes in Noise Pilot, which largely come in five major categories. To embed the design principle of supporting users' existing conceptual models, these nodes are meant to frame actions in the interface as image manipulations. Nodes that interact with the diffusion model for producing a single step of noise estimation should be thought of as producing a noise image, and not have machine learning specific parameterizations.

Image Processing. These nodes implement image processing operations such as linear interpolation, flipping images, gaussian blurring, and applying masks. There are also a set of base mathematical operations like add, subtract, divide, and multiply, which functionally act like image processing nodes. These represent the core set of actions that a user can take on an image in the interface.

Data. This type of node allows the user to manipulate or input data into the interface. The three main data types supported by Noise Pilot are images, scalars, and strings. Images generally represent final output images, noisy image estimates, or images of noise. Scalars (referred to as constants) are used to control certain math operations such as sigma values for gaussian blur or integers that control iteration boundaries. Strings are exclusively used to represent prompts, and are generally only passed as parameters into the diffusion model.

File Operations. Noise Pilot uses these nodes to interact with files. These represent actions like loading image from disk, displaying image outputs in the interface, and a node that allows users to paint simple sketches directly in the node graph.

Denoise Nodes. These are special nodes used within the diffusion process in Noise Pilot. See below for an explanation of how diffusion is implemented in the tool, and Figure 5 for an example of how these nodes connect to implement the process.

Group Nodes. Groups are a special type of node that can contain other nodes. When a group node is placed, the user can double click on it to open the graph "inside" the group. Nodes placed inside the group are executed each time the group is executed, allowing for the creation of reusable functions in Noise Pilot. Data is passed in and out of groups using special Parameter nodes (which are a Data node).

5.3 Iterative Denoise

In Noise Pilot, we represent denoising as special group node which takes parameters necessary to control the iterative process. These include t_{start} which controls which step to start at, t_{end} which controls which step to stop at, an initial image (the starting value

$x_{t_{start}}$), and a prompt. The exposure of the t_{start} and t_{end} parameters allows for users to stop diffusion partway through the process if they want to make edits to the image along the way, and is essential for implementing a number of diffusion-based image generation techniques (See Section 6).

All workspaces start with a default iterative denoise implementation as a group node, which is a direct implementation of DDPM (see Figure 5 for a screenshot of this from the Noise Pilot interface). This group has a few special nodes which help implement the process. *Current Estimate* represents x_t, the current estimate of the image we are generating. *Estimate Noise* estimates the noise in the current estimate given a prompt, which is then removed from the estimate using the *Remove Noise* function. Jointly, these represent the p_θ function. The output of this is fed into *Next Estimate* which represents x_{t-1}, and becomes the current estimate on the next run of the loop.

Implementing the iterative denoise process in the node-based interface allows Noise Pilot to support glass-box abstractions. Users can inspect and manipulate the process of diffusion by inserting operations at different parts of the process. In Section 6, we demonstrate 6 examples of how this can be done and report on the ways that users utilized this feature in the results of our study.

5.4 Implementation

Noise Pilot is implemented as a distributed system consisting of 3 primary modules. See Figure 4 for a complete system diagram.

The user interacts with a frontend called the *Noise Pilot Editor* (described in Section 5.1) through their web browser. Desktop and tablet versions of the interface are supported. The editor is implemented in Typescript and built with React, making heavy use of the react-flow library to implement node interactions. When the workspace is run, the editor turns the workspace into a JSON object that contains the nodes, their parameters, any edges, and the random seed of the run.

This JSON representation is sent to the *Graph Interpreter*, which executes the graph by evaluating each node with a simple execution model. First, a starting node is selected. If any edges in the edge list point to the selected node (as inputs), the nodes connected to those edges must be executed first. Once all of a nodes dependencies have been executed, the PyTorch function associated with the selected node is executed. The results are stored in a lookup table so that nodes executed later can find the results of a node's execution if needed. This process is repeated until all nodes in the graph have been executed. Some nodes send messages back to the editor, such as the previously described *Display Image* node which visualizes an image in the frontend.

Some of the nodes associated with iterative denoise, such as *Estimate Noise*, need to make a call into the diffusion model. These are done by sending a message to the *Diffusion Backend*, which is where the diffusion model is loaded. We use the `DeepFloyd/IF-I-L-v1.0` model due to its tradeoff between relatively low GPU memory requirements (around 20GB) and speed (around 200ms per noise prediction). We also implement classifier free guidance (CFG) [28] to elicit better outputs from `DeepFloyd`.

Figure 6: Left: Outputs from the Visual Anagrams technique. Right: Outputs from the Factorized Diffusion technique. These were made with the following prompts (a) `an oil painting of people sitting around a campire.` (b) `an oil painting of an old man.` (c) `a lithograph of a waterfall.` (d) `a lithograph of a skull`

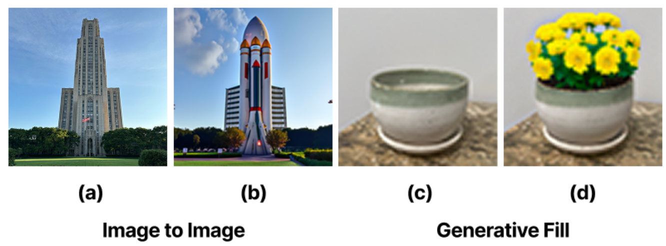

Figure 7: Left: Outputs from the Image to Image translation technique. Right: Outputs from the Generative Fill technique. These were made with the following prompts: (left) `a photo of a rocket ship.` (right) `a flower pot with yellow flowers.`

6 Tool Demonstration

We demonstrate Noise Pilot's capability to support a range of image generation techniques by implementing four existing techniques from the computer vision literature. We also demonstrate that the tool can be used to develop *novel techniques* through two examples developed by the authors and test users while creating the tool. We focus on techniques in this section that would be difficult or **impossible to create by prompting image models** alone to demonstrate the capabilities of a composable workflow based approach.

Algorithms from the computer vision literature were converted directly into workspaces by analyzing the way that they generate images using DDPM. Our own techniques were discovered while prototyping features of the interface, after gaining a better intuition about what kind of operations Noise Pilot supports.

Visual Anagrams. This technique developed by Geng et al. [22] is a method for generating optical illusions from diffusion models. While their paper shows a number of examples of illusions that can be created, we chose to demonstrate the example of *generate an image with two meanings, one of which can only be revealed when the image is viewed upside down*, due to the ease with which this can be viewed on a page (See Figure 6, Left for an example).

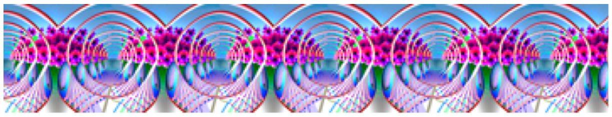

Figure 8: Example of creating a repeating image in the x direction with two prompts: `a vase full of flowers sitting on a table` and `a spiral pattern`.

This illusion is generated by estimating the noise in an image with one prompt, and then estimating the noise in a flipped version of the image with a different prompt. The noise is then flipped back upright and blended with the first noise, before removing noise from the image estimate. See Figure 9 for a screenshot of this technique implemented in the Noise Pilot interface.

Factorized Diffusion. Another technique from Geng et al. [21] allow for the creation of hybrid images, which are images that have two meanings based on how they are percieved or interpreted. A classic example a hybrid image from Oliva et al. [49] is one which contains the high-frequency data of one image, and the low-frequency data in another. Geng implements this with diffusion by predicting noise twice, blurring one, and summing the results.

We implement this example from Geng in Noise Pilot. See Figure 6, Right for an example. The prompt generated at lower frequencies only appears when the image is viewed from a distance, which we've scaled down to simulate. See Figure 15 of the Appendix for a screenshot of this technique implemented in the Noise Pilot interface.

Image to Image Translation. Image to Image Translation as an image generation technique has been long studied in the field of Computer Vision [32, 74]. This technique uses an image as a starting point, and some type of generative process (such as GAN or Diffusion) to change it into a new image that still presents properties of the original. We look to the SDEdit [44] technique as an example of how to implement this technique using iterative diffusion.

This technique works by injecting noise into a starter image, and then running the diffusion loop starting at some $t < T$. This partially diffuses the image and fills in details with the new prompt. See Figure 7 Left for an example output generated with Noise Pilot. See Figure 16 of the Appendix for a screenshot of this technique implemented in the Noise Pilot interface.

Generative Fill. Generative Fill is a technique to fill in blank parts of an image with content based the surroundings. A generative method for performing this operation is often referred to as Inpainting [73]. Inpainting has been implemented for DDPM by Lugmayr's RePaint [42] project.

This technique works by only diffusing the part of an image defined by a mask. Areas outside of the mask have noise injected into them each step of the diffusion process so the new pixels generated inside of the mask can blend with the area outside. See Figure 7, Right for an example generated with Noise Pilot. See Figure 17 of the Appendix for a screenshot of this technique implemented in the Noise Pilot interface.

Tiling Images. A new technique for generating images that tile in the X or Y directions was created during the development of Noise Pilot. This technique works by producing a noise estimate with one prompt, and then rolling the image (an operation that shifts an image in one direction and wraps it back around) before producing another estimate with a different prompt. This creates an image that tiles in the direction of the roll, while being able to represent multiple prompts in the final output. See Figure 8 for an example. See Figure 18 in the Appendix for a screenshot of this technique implemented in the Noise Pilot inteface.

7 Deployment Study

We wanted to understand how real artists would use Noise Pilot for authentic art-making processes. We conducted an extended (2-week) user study with 9 expert participants (creative practitioners with an active computational art practice). This extended study offered several advantages over a shorter lab study, giving participants the time and flexibility to pursue authentic creative explorations, addressing "the novelty effect" (wherein user perceptions of AI systems' usefulness can temper over time [41]), and allowing users to familiarize themselves with the system at a self-directed pace.

7.1 Participants

To recruit participants with expert-level experience and passion for computational art, we shared the study with communities for generative and procedural art-making. After screening for participants with technical and artistic skills, we recruited 10 individuals. We discarded data from 1 participant who was unable to fully complete the study, for a total of n=9 participants.

7.2 Protocol

The study consisted of a 60-minute entrance interview, 10 hours of tool use over the course of 2 weeks, and a 60-minute exit interview. Participants received a $200 USD gift card for completing the study. Interviews were conducted and transcribed using Zoom. The first or second author attended each interview, and reviewed recordings of interviews they were not present for. Our protocol was approved by our institutional review board.

The entrance interview consisted of questions to understand the participant's prior experiences and creative practice, and onboarding, which included:

- providing participants with a document containing a link to the system, their login information, and links to a documentation website and tutorial videos,
- watching a video explaining the basics of diffusion by Steve Seitz[1] so that all participants would share a baseline understanding of the diffusion process
- a live demonstration of Noise Pilot — we demonstrated reimplementing "Visual Anagrams" (see Section 6), and
- a walkthrough tutorial — we guided each participant through implementing image-to-image in Noise Pilot.

Participants were also given access to 5 template workspaces (standard diffusion, visual anagrams, factorized diffusion, image-to-image, and generative fill from Section 6) that they could base their creations off of, as well as a tutorial video walking through how to build each one from scratch. Participants are able to clone the templates to use them as starting points for their own work.

[1]https://www.youtube.com/watch?v=lyodbLwb2lY

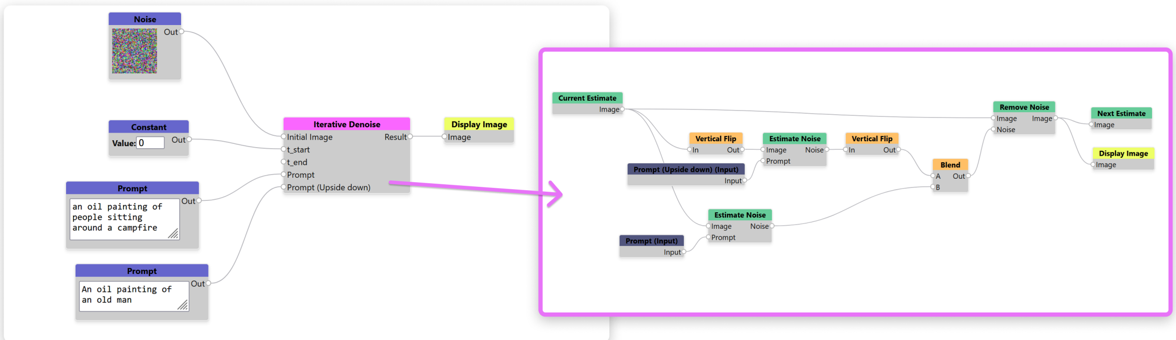

Figure 9: Example of the visual anagrams workspace as it appears in Noise Pilot. The callout box represents the custom Iterative Denoise node necessary to implement this technique. Current estimates are denoised twice with different prompts each. One of the noise estimates is flipped to embed the second prompt as being visible when the image is viewed upside down.

Participants were then asked to use the tool for ~10 hours over the course of the next two weeks — we suggested 30-45 minutes of tool use per day, but participants were given the freedom to decide the length and structure of their use sessions.

Participants were asked to submit a reflection form with their favorite art pieces made during the session of tool use, a link to the workspaces they used to make the images, and asked them to fill out the following questions:

- Please use this space to share any notes you have about the files you shared above.
- Please feel invited to use this space to share any thoughts, notes, comments, questions, or ideas you might have.

Our exit interview protocol first invited participants to share their open-ended responses to the system. We then used screen-sharing to show each participant the workspaces and artifacts they created in Noise Pilot, to contextually ground the rest of the interview. This took place after our initial rounds of data analysis, allowing participants to play an active role in the interpretation of the data, as we detail below.

7.3 Data Collection and Analysis

We collected and analyzed data on participants' use of Noise Pilot: as observed by reviewing their workspaces, usage data, and resulting artifacts, and as described and contextualized in their interviews and form submissions.

Seeking an understanding authentically grounded in our expert participants' diverse creative processes, we took an iterative, inductive, and constructivist approach to the thematic analysis of our qualitative data – where both researcher and participant take an active role in interpreting the data and constructing understanding from it [11].

We began analysis during the 2-week deployment study, continually reviewing participants' workspaces, artifacts, and submissions to the reflection form, and developing a set of initial codes. Before each participant's exit interview, the first and second authors compiled their form submissions, artifacts, and screenshots of their workspaces into a digital whiteboard application, using post-it notes to annotate with initial codes and observations.

During the exit interview, we invited participants to discuss their process. We shared our initial interpretations, allowing them to challenge, confirm, or elaborate upon them. The first and second authors met before and after each interview, and 3 additional times throughout the analysis, to iteratively refine our approach.

We used the refined codes to analyze each exit interview transcript, then completed a second pass over all participant materials. We duplicated digital post-its to another board to support clustering by topic, ultimately finalizing the themes that guide the subsequent presentation and discussion of our results.

8 Results

Participants created 167 workspaces during the study, ranging from 2 workspaces (P3) to 29 workspaces (P6, P7, P9). Over the two week period, participants generated 7295 images. 2808 of those were generated in a single workspace by P8.

Participants made 17815 edits to workspaces during the study, 9341 of which we classify as *major* changes. We defined a major change as any that would affect the output image, so this excludes simply moving nodes or organizing their workspace. We further classify their actions into three categories. We classify changes to *prompt* or *constant* nodes as **Prompting** edits — an action intended to alter image output by tweaking input parameters. We classify any graph modifications (adding nodes, removing nodes, adding edges between nodes, etc) performed outside of an iterative denoise as **Diffusion Pipeline** edits — actions that change the way the image is generated using combinations of image manipulation operations, but not one that changes how denoise fundamentally happens. Any graph modifications made inside of an iterative denoise group we classify as a **Diffusion Customization** edit — one that changes the way that diffusion occurs. We summarize these edits for each user in Figure 11.

Those with prior experience with AI models remarked that the system felt "familiar" (P5) and "intuitive" (P9) enough to adapt to support their existing creative processes and goals, even if (like P4) they had no prior experience using such a node-based programming interface:

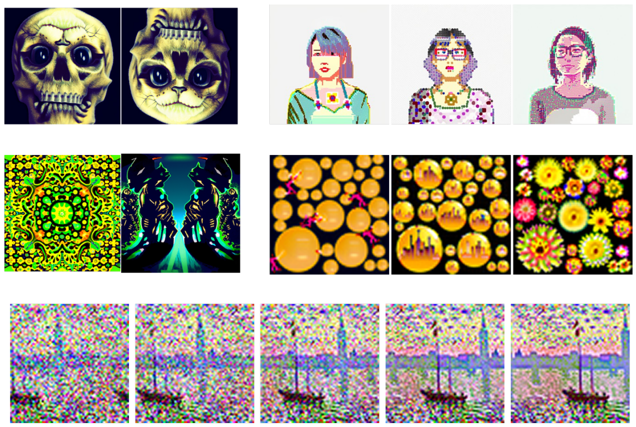

Figure 10: A sample of some of the art our participants generated. P5 (top, left) used the visual anagram template to prompt for images incorporating animals + skulls, a recurring theme in their work (original output at left; we show it horizontally flipped at right). Meanwhile, P3 (middle, left) adapted the template to explore visual symmetries. P2 (top, right) used the image-to-image translation template to generate pixelated character avatars. P8 (middle, right) imported their own art into Noise Pilot and create a series of variations. P7 (bottom) create their own custom iterative denoise process to generate a custom effect, which led to an animating boat as the image diffused.

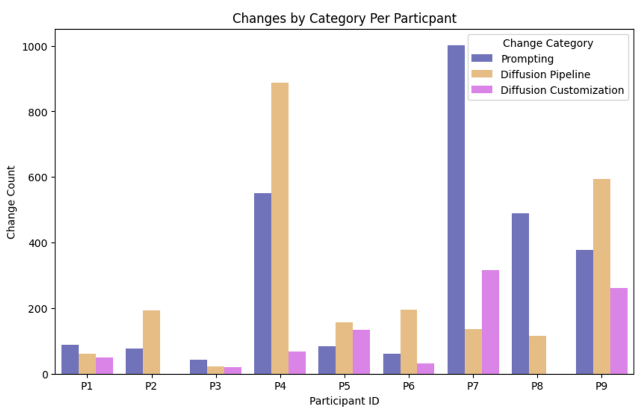

Figure 11: A plot of the number of changes we observed users make during the study, categorized by which level of depth the change was made at. Prompting changes represent the number of times users changed parameters in their workspace (either prompts or constants). Pipeline changes represent actions taken to build pipelines, but not modify the diffusion process. Customization actions represent actions taken to modify the way that iterative denoise happens for their workspace.

> I definitely used my existing generative AI infused workflow. I was using it like I would use any other tool. (P4)

Our participants came from a variety of computational art backgrounds; three (P2, P6, P8) had little to no experience using AI in their creative practice, instead using creative coding tools like Processing to make procedural generative art. All three were able to bring their existing skills and interests into Noise Pilot, and explicitly noted that they found it intuitive to learn and use.

P6 described Noise Pilot's low thresholds, discussing how they were able to successfully create and use a custom workflow in their first use session:

> "It actually wasn't that hard... That's like, the beauty of this tool." (P6)

Next, we characterize the different ways participants manipulated diffusion in Noise Pilot, across the three levels of increasing manipulative depth: by *prompting* and parameter tweaking, by building diffusion *pipelines*, and by modifying the diffusion process itself with *diffusion customization.*

8.1 Supporting High-Level Explorations with Prompting

First, we discuss how participants manipulated diffusion at the highest level of abstraction: by iteratively tweaking prompts and parameters, including prompting the provided template workspaces. (As mentioned in Section 7, we provided participants with five premade template workspaces, each containing a different diffusion-based technique, implemented as a reusable node-based program.)

All users performed some amount of prompting and tweaking of parameters, for a total of 3183 such changes across users. Some users preferred this modality of interaction, using it more than other forms (P2, P3, P7, P8). This is to be expected; manipulating diffusion at the high-level of abstraction afforded by prompting is both straightforward and immediately available in Noise Pilot.

Manipulating diffusion at the prompting "level" does support a wide breadth of exploration and expressivity, as noted by P8:

> Even just within the templates you guys provide ...there's just a lot of parameters, a lot of degrees of freedom. You could do it forever. (P8)

Figure 10 shows some art produced by our participants using the provided templates. P5 created Visual Anagrams incorporating animals and skulls, a recurring theme in their work. Meanwhile, P3 adapted the Visual Anagram template to explore visual symmetries.

While some participants were satisfied with exploring at the prompting level, they could sense the "depth" of manipulation available in the system, which motivated some to explore further. After completing many extensive prompting explorations, P8 became interested in introducing more complexity and "depth" into their process:

> "I was just sort of scratching the surface, and that might have to do with, you know, the depth of the tool. [...] I sort of felt obligated to branch out, I got to see what else is here." (P8)

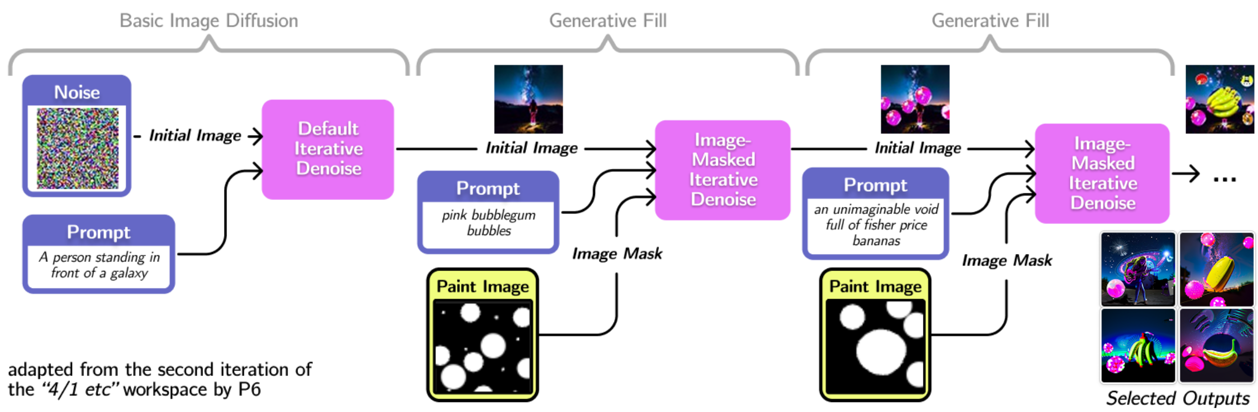

Figure 12: P6 first created this workspace following along to our video tutorial for reimplementing "generative fill" — which entails customizing the diffusion procedure to generate within a given image mask. He then realized he could "chain" diffusion and generative fill procedures together to additively shape the resulting outputs. After achieving an interesting result, P6 would duplicate the workspace to preserve its structure, then continue to add changes in a separate copy. This second iteration layers two "generative fills" onto an initial generation; his final version incorporates 10 additional layers by repeating this structure and choosing a new prompt for each successive layer.

8.2 Supporting Flexible Workflow Composition with Reusable "Diffusion Pipelines"

Participants (P1, P3, P4, P9) noted that they often import and export between different (AI and image-editing) tools in their existing creative practices with AI, effectively composing different tools together into a bespoke creative workflow. Noise Pilot users were able to do this kind of workflow authoring within the tool environment itself; across all users, we identified 3794 workspace actions representing this kind of diffusion "pipeline" manipulation.

Feeding AI-generated images back into models (either from the same or from different models) is a recurring interest in P9's practice:

> "I love to take stuff that I generate and then feed it back in... this tool lets me kind of like, weave that together in one shot. I don't have to make an image, and then copy it and paste it over somewhere else [...] How delightful it was to be able to get that pipeline set up in one go." (P9)

An artist might build a diffusion pipeline to feed the output of one or more diffusion processes into another, or to apply image processing (e.g., transformations, color channel manipulation, or masking) before or after diffusion. They might also use pipelines to automate a repetitive workflow.

P4 used Noise Pilot to implement what they titled a "Pixel Art Factory" that generates 8 versions of a single prompt with each run (e.g., "vintage 1995 graphic art of a tiny red mushroom with blue spots, white background"); each output is then sent to an upscaler node with the prompt, "high-definition pixel art" (as shown in Figure 13):

> "Here I'm using the first part to establish compositions that I like, and then using the second part to establish more of a stylistic style that I like." (P4)

They used red mushrooms as their subject while iteratively developing the pipeline and prompt. Once built, she reused this "factory" with minor prompt modifications to generate mushrooms in various colors, then butterflies (see Figure 13).

Diffusion pipelines allowed artists to build bespoke tooling — reusable artifacts that enact a process. These pipelines flexibly incorporate diffusion as a duplicable tool in their process — but Noise Pilot also offers them the option to "break into" that tool and customize its behavior, which we report on next.

8.3 Supporting a Depth of Manipulation with Diffusion Customization

The diffusion process is implemented within Noise Pilot as the Iterative Denoise Node. Users can change the diffusion process by editing its implementation. We observed 2364 "diffusion customization" changes (compare this to 3183 prompting changes, and 3794 pipeline changes; see the pink bars in Figure 11). 7 out of 9 participants performed some kind of diffusion customization, with 2 participants (P5 and P7) customizing diffusion as their second most type of activity. As the "deepest" level of control available, diffusion customization affords uniquely powerful expressive capabilities — but wielding such granular control is not necessarily easily accessible to all users. Additionally, once a user builds a custom diffusion process, they can immediately reuse it: incorporating it as a tool in larger diffusion pipelines, and iteratively prompting with it. We might expect fewer changes in this category for these reasons.

P9 used diffusion customization to "inject" photography directly into the denoising process, as part of a goal that he has pursued in prior work:

> "Something I've been wrestling with with AI for a while is trying to get more of my original creativity injected into that process." (P9)

Figure 14 depicts the custom iterative denoise node P9 built into a workspace titled "subtle blend." P9 modified the node to use two

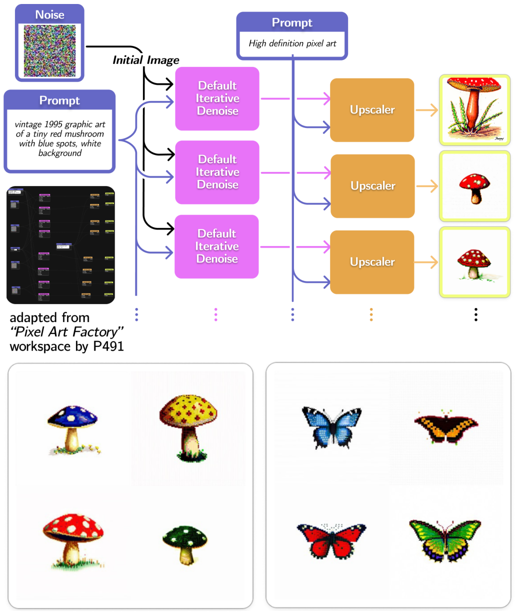

Figure 13: P4's "Pixel Art Factory" workspace generates 8 images from a single prompt at a time, then stylistically upscales each as "high-definition pixel art." They returned to this workspace multiple times throughout the study, reusing it to make stylistically consistent pixel art of various subjects.

additional inputs: an image that will get blended into each step of the diffusion process using linear interpolation, and an Alpha parameter that controls the "strength" of the blended image's influence. They discovered that using a high alpha generates an image stylistically influenced by the input image with more "subtlety" than afforded by the image-to-image approach, and that it has the additional stylistic effect of creating a soft "blended" look. P9 used this workspace together with a photograph of a building to produce a series of dreamy, stylistically consistent outputs (Figure 14).

P7 explored diffusion customization in their workspace, "Rolling Noise and Mask" (see Figure 19 in the Appendix.) Rather than pursuing a particular goal or vision, P7 gradually added new components to the denoise loop, iteratively testing how each addition influenced the output. As P7 discovered interesting new possibilities, they let this shape the ongoing exploration. For example, he found that adding "Roll Image" nodes to the iterative denoise process will shift the image with each diffusion step. As the system generated and displayed each intermediate output, P7 was surprised to find that it appeared to animate.

Guided by this discovery, P7 used this workspace to combine a photo of sailboats and art-history inspired prompts and create image series where boats appear to move across the water, as the image diffuses from pure noise (see Figure 10).

> "At this point, I had had some more familiarity with the tool. So it was like, things were making more sense and then, like, I tried something and like, oh, I want to roll the image and see like... Wouldn't that just make it move? And then I did it. It's like, oh yeah, it is moving! Holy crap. That's cool." (P7)

8.4 Interactions Across Three Levels of Interaction Depth

Overall, participants made Pipeline level edits the most (3794), Prompting changes second (3183), and Diffusion Customization the least (2364). Noise Pilot made three levels of interaction "depth" accessible to users — beyond interacting within each level, participants made use of the affordances of being able to move freely between them.

8.4.1 Displaying intermediate outputs of the diffusion process creates useful visibility. Noise Pilot creates *visibility* into what is typically a black-box process, which participants found impactful across all three levels of interaction depth.

As seen in artists' use of other generative AI systems (e.g., [2]), participants used iterative prompting to "probe" and sensemake the diffusion model: to better understand its capabilities, and to leverage those capabilities for future actions.

We found that allowing users to view each frame of the diffusion process provided insightful visual feedback, supporting this creative intuition building:

> "From, like, a creative point of view, it was really interesting to see how it does what it's doing, like, watching as it iterates over and spits images out... It was very helpful because you can see, because it's not always immediate that something has gone sideways. ...yeah, that feedback was super useful." (P7)

Participants used display nodes as instruments for probing for further visual insight into the diffusion process throughout their workflows:

> "It was definitely useful to watch... sometimes I would even disconnect a piece or add another display image partway through to get, like, intermediate results." (P9)

8.4.2 Lower-level compositions become reusable units of higher-level tooling. Artifacts of participants' explorations became reusable and iterable custom-tooling that they could incorporate into their pipelines. Beyond creating custom diffusion processes, participants treated groups of nodes and entire workspaces as reusable units of functionality.

While following one of our video tutorials for implementing generative fill, P6 realized they could "chain" diffusion and generative fill procedures together to additively shape the resulting outputs. After achieving an interesting result, P6 would duplicate the workspace to preserve its structure, then continue to add layers and modifications to the new copy. Figure 12 is the second iteration of this pipeline; it layers two "generative fills" onto an initial generation. P6 wrote that their final version incorporates 10 additional layers by repeating this structure and choosing a new prompt for each successive layer. From their form responses, where they submitted comments on each workspace in this "series":

> "I wanted generate an image, and then generate a completely new image within masks — to sort of tease

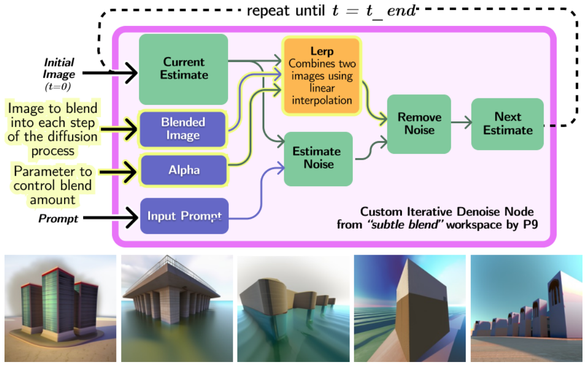

Figure 14: P9 built a customized diffusion process (above) that subtly blends an image into each step of the diffusion process, and used it to generate a stylistically consistent series of outputs (below). Components that P9 added to the default diffusion process are highlighted in yellow.

> out the way that the medium works in different ways… I'm loving being able to get very granular in how I can start to create an ultra-layered deep-fryable image. […] I also feel like this result could not easily be prompted for." (P6)

Artists who customized the diffusion process experienced a sense of the "materiality" of diffusion by manipulating it at this low-level of "depth." While participants appreciated the power of diffusion customization, the struggle inherent at this level of "depth" led participants to work their way towards this level gradually, with some returning to favor interaction at higher-levels of abstraction.

> "It was just a variety of unfamiliarities, and so I wanted to start really simple, and I like to work through things sort of methodically." (P8)

> "I find that I really need to force myself to use some of the novel features of Noise Pilot. Prompting alone is still king! Especially …where I haven't had much experience with the model in the first place…" (P4)

Participants were motivated to engage at the lower-level of manipulative "depth" when they conceived of a "custom diffusion" idea or effect that they could not achieve through prompting or by combining existing nodes. However, successfully implementing a particular vision demanded more granular effort:

> "I had a little bit of difficulty in implementing some of the ideas that I had… I was kind of starting to try to explore some more prompt travel, but having some difficulty with it…" (P5)

Templates served as useful starting points, making exploration of custom diffusion processes more accessible. Participants often started by choosing a template to duplicate (e.g., the Visual Anagram or Factorized Diffusion implementations) then tweaked it to make it their own. Still, they experienced failures and unexpected outcomes:

> "One of the things I was doing was trying to create distinct images in the red, green, and blue channels of images and so it was like, I would build what I *thought* would achieve that, try it, and then it wouldn't work. …it's hard… I don't know if it's even possible to do." (P7)

This struggle inspired P7 to abandon goal-directed exploration in favor of more open-ended, undirected "play":

> "I just want to flip switches and connect whatever, not even really think about the rationale behind it. I just want to see what happens and break things …If I want to break stuff, it's not stressful at all." (P7)

This characterizes how P7 achieved their generative sailboat "animation" (see Figure 19 in the Appendix). Embracing breakage and "failure" came at the cost of interpretability — a struggle to fully articulate *how* their implementation works, and a sense that they had not necessarily executed it "correctly." However, this undirected "playing" with diffusion led them towards moments of creative serendipity, where they identified a surprising new direction they did not previously conceive as possible.

P9 described how, although the "subtle blend" tooling they built (see Figure 14) was only marginally different from features they'd used in other AI applications, they experienced a unique sense of manual, material engagement developing the functionality themselves.

> "It's kind of the same thing they do with like, Leonardo or anything else, but it did feel kind of different… It just felt like there was some… tangible process around that I was able to, you know, work with. […] 'Oh, what happens if I tamper with this variable here, or drop this in here?' You're like, building, or sculpting, you don't quite know where it's going to go, but you'll know it when you see it. …more like crafting and building something, I feel, than just purely typing and then hitting generate a ton of times. (P9)

9 Discussion

Noise Pilot gave artists ways to engage with image diffusion across three levels of abstraction depth: supporting high-level prompt exploration, flexible workflow composition, and direct manipulation of the diffusion process itself. Presenting diffusion within a multi-layer interface where the process can be probed, shaped, and assembled with other image operations invited artists to engage with its *materiality.*

Here we situate our findings within literature on the impact of materiality and multi-layer interfaces on creative behavior, highlighting how future tool designers can navigate the opportunities and tradeoffs of these dynamics in creative practice with generative AI.

9.1 Engaging With the Materiality of Diffusion: Opportunities and Trade-Offs

> "I engage with the tool totally differently than with a UI where I type in a prompt, I get my image and I click a button that says, hey, upscale this one…. this

> UI changed the way that I engaged with it, and I definitely would have never thought to do this." (P9)

In designing Noise Pilot as an interface for art-making with image diffusion, we sought to *enable more reflective conversation between artist and the material* than is typically found in prompt-centered generative AI interaction design, and to characterize the role this plays in artists' creative engagement with the diffusion medium.

Prior work has identified how *materials* — "the elements of the task environment that can be shaped, sensed, probed and assembled" [8, 67] influence the possibilities that practitioners see, and choose to realize, in creative activity. In seeking to support more meaningful art-making experiences with the *materiality* of 3D fabrication technology, Devendorf & Ryokai [14] drew upon Ingold's [31] conceptualization of "non-hylomorphic" making to support more open-ended, exploratory making.

Devendorf & Ryokai noted a trend in 3D-printer interaction design: a naively "hylomorphic" view that treats materials as passive, and reduces art-making to the task of imposing some preconceived vision of a final form onto that material.

We argue that prompt-centered generative AI interaction design often embodies this same hylomorphic expectation: the artist articulates their vision as a text prompt; the diffusion model's role is to produce an image that "accurately" realizes the artist's vision.

In contrast, a "non-hylomorphic" view of making understands creation as a reflective practice [57, 58]: a *correspondence* between artists and the "active materials" they work with "in anticipation of what might emerge" — the final form as something that is *not* predetermined, but gradually *shaped*, as the maker takes actions that probe and respond to the material's affordances [14, 31].

Why does this matter in the design of digital CSTs for apparently "immaterial" digital mediums, like image diffusion?

Active, reflective "material" correspondence requires that the artist is able to take *actions* to gain insight into the medium's properties and capabilities. Tricaud & Beaudouin-Lafon [67] describe these *epistemic actions* as fundamental to creative behavior, and necessary for building confidence in a creative medium. They call for digital creativity support tools to "reclaim" materiality in otherwise "immaterial" software environments by making instruments available for perceiving the computational material's properties and capabilities, while still retaining algorithmic expressivity.

Existing tools for interacting with generative AI obscure their processes within "black-box" abstractions, or with anthropomorphism, and provide prompting as the central method for *indirectly* manipulating them.

Users of such tools are not likely to understand diffusion as an iterative process, nor see how this iteration evolves over time, nor see how their actions create changes to inputs, that then map to changes in the underlying process, and its outputs.

This normatively exerts the assumption that end-users would not, or could not, be interested in manipulating AI models at these lower levels of abstraction. As articulated by P4:

> "...the diffusion process in so many online generators gets obscured now or just like, not shown to the user, because like, 'why would the user care[...]?' But it is a very cool process to learn about and helps you understand the models better. And I think by virtue of doing that, you use the tools in more imaginative ways." (P4)

By designing for *glass-box abstractions*, Noise Pilot users are able to gather insight into the nature of diffusion as a manipulable image *material*. They can take action to discover how diffusion can be shaped, probed, assembled and instrumented together with other active image materials [14]. The impact of this material visibiilty and access emerged throughout our findings, as participants compared their experiences to "sculpting" (P9), or noted the "texture" (P6) of diffusion; as they described using the instruments in Noise Pilot to support open-ended, epistemic exploration (P4, P6, P7, P8, P9) — creating to play with, discover, and build confidence in diffusion as a *medium*, and allowing those explorations to shape their creativity in turn.

Noise Pilot participants also utilized pipelines (e.g., P6 in Figure 12, P4 in Figure 13) and built custom diffusion procedures (e.g., Figure 14, 19 in the Appendix) to create what Tricaud & Beaudouin-Lafon term *epistemic artifacts* — persistent, reusable externalizations of knowledge that enrich the medium with new possibilities for action. Such artifacts are readily repurposed as shareable tools — creating a foundation for creative community where "power-users" with interest in tool-building can share artifacts to support other users' processes.

However, engaging at this depth is not without its tradeoffs. Users encountered more "failures" here — struggling with the granular decision-making required to manipulate these primitive "materials" towards a successful execution of higher-level ideas, and finding that their changes could quickly steer the diffusion process towards unexpected outputs, e.g., heavily distorted or noisy images.

Any sufficiently expressive medium creates opportunities for failure: prior work [34] describe the importance of supporting failure in CSTs to create opportunities for growth, experimentation, and surprise. Rather than prescriptively abstracting the possibility of "failure" away, Noise Pilot gave artists the *option* to more directly engage in a reflective struggle with the materiality of diffusion — a struggle that some artists chose to embrace (e.g., creating *intentionally* glitchy, noisy and diffusion "textured" pieces). However, only some participants took this option: 2 out of 9 participants did not make customized diffusion processes. Next, we reflect upon designing generative AI CSTs to support users' power to *choose*, and *refuse*, to engage at particular depths of abstraction depending on their interests and needs.

9.2 Empowering "Vertical Movement" in Generative AI CST Design

Noise Pilot is a multi-layer interface where users freely moved between levels of abstraction in their interactions with image diffusion. We place our findings in conversation with the body of critical CST work arguing that, rather than designing for *alignment* with creative needs, CSTs should support *flexibility* as practitioners' needs shift and change. In choosing what to abstract, tool designers enact *power-over* creative practitioners, constructing the space of possible actions that practitioners can see and manipulate within the tool [39]. While practitioners benefit when the system's set of chosen abstractions *align* with their needs and goals, abstraction can hinder the ways artists learn and manipulate: through manual engagement with materials [38].

While engaging with diffusion at a material level provides a depth of opportunities not accessible with prompting interactions alone, it also comes with tradeoffs that not all users want or need constantly. We observed that the "power users" who used diffusion customization the most (P7, P9) were among the users who used Noise Pilot the most during the study. This usage patterns align with Shneiderman's description of how a multi-layered interface "... allows novices to begin a first layer and move up as their experience increases and needs require" [59]. Only more experienced users reached for deeper levels of control when they felt they needed it — and when they did, they were able to then use and reuse these custom tools in higher-level pipelines.

Rather than attempting to choose and prescribe a single "correct" set of abstractions for interacting with generative AI models, we can shift power away from tool designers and towards practitioners, by designing interfaces that enable them to freely navigate up and down levels of abstraction — what Li et al. call "*vertical movement*" [39].

Noise Pilot users flexibly moved through the three levels of manipulative depth accessible in Noise Pilot, depending on their creative needs, experience, and interests. We follow Li et al. in insisting that software designers offer artists the flexibility to choose, manipulate, disrupt, combine, and even reject tools entirely [39].

By enabling through material visibility, manipulation and appropriation, Noise Pilot also empowered artists to show *us*, the system designers and researchers, new capabilities and qualities to this medium that *we* would not have understood otherwise. We encourage *artistic-support tool* researchers to further explore how we might design interfaces that shift power in technical design, and in HCI research, towards artists. We believe that embracing *materiality* and medium-specificity is a promising approach: by allowing artists to leverage that which is "unique in the nature of the medium" [24] — the medium-specific characteristics of particular AI models and algorithms — we can empower artists to directly contribute to shaping way we understand and build with these technologies.

10 Limitations and Future Work

Noise Pilot supported a maximum image resolution of 256x256 pixels. While some participants (P2, P4, P8) explicitly noted feeling creatively intrigued by this constraint (motivating pixel-art style explorations, for example), two participants (P1, P3) said this hindered them from achieving styles that they wanted to pursue (e.g., photorealism).

Moreover, Noise Pilot does not support manipulating a generative model at the *training* level — gathering and iteratively curating a dataset to condition a model to output similar images. This is a dimension of control over diffusion that P1 and P3 frequently use in their creative processes with "LoRAs" [29].

These technical limitations relate to our choice of model (DeepFloyd's 64x64 base module and separate 256x256 upscaler), which together offered capabilities appropriate for the focus of this study. Noise Pilot would, however, readily support swapping in another model or offering a choice of models in the future. How can changing the underlying model, while keeping the interface the same, affect the materiality of the system — and vice versa? Future work might attempt to better disentangle this: which medium-specific characteristics are being enacted by model design, by interface design, or by sociocultural factors outside of the designer's control?

11 Conclusion

New methods for AI art making can be explored by opening up the black box of diffusion models and building tools that engage with algorithms at deeper levels. To explore this, we built Noise Pilot, a node based interface for generating images with diffusion by implementing the DDPM algorithm as nodes along side image processing nodes which can be used to manipulate how images are created. We show that the tool is effective at implementing state-of-the-art diffusion based techniques, as well as new custom techniques. Our two week-deployment with 9 participants found that artists engaging with our tool across three levels of depth. We connect these findings to literature on materiality and artistic empowerment, suggesting that we might better empower artists interactions with black-box systems by designing to support more material-like manipulation.

References

[1] Cycling '74. 2024. *What is Max?* https://cycling74.com/products/max
[2] Shm Garanganao Almeda, J.D. Zamfirescu-Pereira, Kyu Won Kim, Pradeep Mani Rathnam, and Bjoern Hartmann. 2024. Prompting for Discovery: Flexible Sense-Making for AI Art-Making with Dreamsheets. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3613904.3642858
[3] Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization Effects of Large Language Models on Human Creative Ideation. In *Creativity and Cognition*. ACM, Chicago IL USA, 413–425. doi:10.1145/3635636.3656204
[4] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. 2023. Spellburst: A Node-based Interface for Exploratory Creative Coding with Natural Language Prompts. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–22. doi:10.1145/3586183.3606719
[5] Robert McNeel & Associates. 2024. *Grasshopper - Rhino - Features*. https://www.rhino3d.com/features/#grasshopper
[6] AUTOMATIC1111. 2025. *AUTOMATIC1111/stable-diffusion-webui: Stable Diffusion web UI*. https://github.com/AUTOMATIC1111/stable-diffusion-webui
[7] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-Image Generation through Interactive Prompt Exploration with Large Language Models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3586183.3606725
[8] Tone Bratteteig. 2010. A Matter of Digital Materiality. In *Exploring Digital Design: Multi-Disciplinary Design Practices*, Ina Wagner, Tone Bratteteig, and Dagny Stuedahl (Eds.). Springer, London, 147–169. doi:10.1007/978-1-84996-223-0_5
[9] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. doi:10.48550/arXiv.2211.09800 arXiv:2211.09800 [cs].
[10] Minsuk Chang, Stefania Druga, Alexander J. Fiannaca, Pedro Vergani, Chinmay Kulkarni, Carrie J Cai, and Michael Terry. 2023. The Prompt Artists. In *Proceedings of the 15th Conference on Creativity and Cognition (C&C '23)*. Association for Computing Machinery, New York, NY, USA, 75–87. doi:10.1145/3591196.3593515
[11] Kathy Charmaz. 2014. *Constructing Grounded Theory*. SAGE. Google-Books-ID: v_GGAwAAQBAJ.
[12] DaEun Choi, Sumin Hong, Jeongeon Park, John Joon Young Chung, and Juho Kim. 2024. CreativeConnect: Supporting Reference Recombination for Graphic Design Ideation with Generative AI. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–25. doi:10.1145/3613904.3642794

[13] John Joon Young Chung and Eytan Adar. 2023. PromptPaint: Steering Text-to-Image Generation Through Paint Medium-like Interactions. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3586183.3606777

[14] Laura Devendorf and Kimiko Ryokai. 2015. Being the Machine: Reconfiguring Agency and Control in Hybrid Fabrication. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 2477–2486. doi:10.1145/2702123.2702547

[15] Jane L. E, Yu-Chun Grace Yen, Isabelle Yan Pan, Grace Lin, Mingyi Li, Hyoungwook Jin, Mengyi Chen, Haijun Xia, and Steven P. Dow. 2024. When to Give Feedback: Exploring Tradeoffs in the Timing of Design Feedback. In *Proceedings of the 16th Conference on Creativity & Cognition (C&C '24)*. Association for Computing Machinery, New York, NY, USA, 292–310. doi:10.1145/3635636.3656183

[16] Gerhard Fischer, Kumiyo Nakakoji, Jonathan Ostwald, Gerry Stahl, and Tamara Sumner. 1993. Embedding Critics in Design Environments. *The Knowledge Engineering Review* 8, 4 (Dec. 1993), 285–307. doi:10.1017/S026988890000031X

[17] Blender Foundation. 2021. *Shader Editor — Blender Manual*. https://docs.blender.org/manual/en/2.81/editors/shader_editor/index.html

[18] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. 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, New York, NY, USA, 1–18. doi:10.1145/3290605.3300619

[19] Jonas Frich, Michael Mose Biskjaer, and Peter Dalsgaard. 2018. Twenty Years of Creativity Research in Human-Computer Interaction: Current State and Future Directions. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. Association for Computing Machinery, New York, NY, USA, 1235–1257. doi:10.1145/3196709.3196732

[20] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 2414–2423. doi:10.1109/CVPR.2016.265

[21] Daniel Geng, Inbum Park, and Andrew Owens. 2024. Factorized Diffusion: Perceptual Illusions by Noise Decomposition. doi:10.48550/arXiv.2404.11615 arXiv:2404.11615.

[22] Daniel Geng, Inbum Park, and Andrew Owens. 2024. Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models. doi:10.48550/arXiv.2311.17919 arXiv:2311.17919.

[23] Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. doi:10.1145/3290605.3300526

[24] Clement Greenberg. 1982. Modernist Painting. In *Modern Art And Modernism*. Routledge. Num Pages: 6.

[25] Yuhan Guo, Hanning Shao, Can Liu, Kai Xu, and Xiaoru Yuan. 2024. PrompTHis: Visualizing the Process and Influence of Prompt Editing during Text-to-Image Creation. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–12. doi:10.1109/TVCG.2024.3408255

[26] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 299–308. doi:10.1145/1166253.1166300

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. doi:10.48550/arXiv.2006.11239 arXiv:2006.11239 [cs].

[28] Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. doi:10.48550/arXiv.2207.12598 arXiv:2207.12598 [cs].

[29] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. doi:10.48550/arXiv.2106.09685 arXiv:2106.09685 [cs].

[30] T. Igoe. 2011. *Making Things Talk: Using Sensors, Networks, and Arduino to See, Hear, and Feel Your World*. O'Reilly Media, Incorporated. https://books.google.com/books?id=14HZK0Pi-mEC

[31] T. Ingold. 2010. The textility of making. *Cambridge Journal of Economics* 34, 1 (Jan. 2010), 91–102. doi:10.1093/cje/bep042

[32] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976. doi:10.1109/CVPR.2017.632 ISSN: 1063-6919.

[33] Jennifer Jacobs, Joel Brandt, Radomír Mech, and Mitchel Resnick. 2018. Extending Manual Drawing Practices with Artist-Centric Programming Tools. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3174164

[34] Joy Kim, Avi Bagla, and Michael S. Bernstein. 2015. Designing Creativity Support Tools for Failure. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (C&C '15)*. Association for Computing Machinery, New York, NY, USA, 157–160. doi:10.1145/2757226.2764542

[35] Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. 2023. Large-scale Text-to-Image Generation Models for Visual Artists' Creative Works. In *Proceedings of the 28th International Conference on Intelligent User Interfaces (IUI '23)*. Association for Computing Machinery, New York, NY, USA, 919–933. doi:10.1145/3581641.3584078

[36] Max Kreminski. 2024. Computational Poetry Is Lost Poetry. In *Proceedings of the Halfway to the Future Symposium*. ACM, Santa Cruz CA USA, 1–4. doi:10.1145/3686169.3686179

[37] Jingyi Li, Joel Brandt, Radomír Mech, Maneesh Agrawala, and Jennifer Jacobs. 2020. Supporting Visual Artists in Programming through Direct Inspection and Control of Program Execution. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. doi:10.1145/3313831.3376765

[38] Jingyi Li, Sonia Hashim, and Jennifer Jacobs. 2021. What We Can Learn From Visual Artists About Software Development. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–14. doi:10.1145/3411764.3445682

[39] Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer. 2023. Beyond the Artifact: Power as a Lens for Creativity Support Tools. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3586183.3606831

[40] Haichuan Lin, Yilin Ye, Jiazhi Xia, and Wei Zeng. 2025. SketchFlex: Facilitating Spatial-Semantic Coherence in Text-to-Image Generation with Region-Based Sketches. doi:10.48550/arXiv.2502.07556 arXiv:2502.07556 [cs].

[41] Tao Long, Katy Ilonka Gero, and Lydia B Chilton. 2024. Not Just Novelty: A Longitudinal Study on Utility and Customization of an AI Workflow. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference (DIS '24)*. Association for Computing Machinery, New York, NY, USA, 782–803. doi:10.1145/3643834.3661587

[42] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11451–11461. doi:10.1109/CVPR52688.2022.01117 ISSN: 2575-7075.

[43] Jiaju Ma, Li-Yi Wei, and Rubaiat Habib Kazi. 2022. A Layered Authoring Tool for Stylized 3D animations. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 383, 14 pages. doi:10.1145/3491102.3501894

[44] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. doi:10.48550/arXiv.2108.01073 arXiv:2108.01073.

[45] Midjourney. 2024. *Midjourney*. https://www.midjourney.com/website

[46] Mark J. Nelson, Swen E. Gaudl, Simon Colton, and Sebastian Deterding. 2018. Curious users of casual creators. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (Malmö, Sweden) *(FDG '18)*. Association for Computing Machinery, New York, NY, USA, Article 61, 6 pages. doi:10.1145/3235765.3235826

[47] NVIDIA. 2024. *NVIDIA Canvas App: AI-Powered Painting*. https://www.nvidia.com/en-us/studio/canvas/

[48] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I Lead, You Help but Only with Enough Details: Understanding User Experience of Co-Creation with Artificial Intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3174223

[49] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. 2006. Hybrid images. *ACM Trans. Graph.* 25, 3 (July 2006), 527–532. doi:10.1145/1141911.1141919

[50] OpenAI. 2024. *ChatGPT*. https://chatgpt.com

[51] OpenAI. 2024. *DALL·E 3*. https://openai.com/index/dall-e-3/

[52] Jonas Oppenlaender. 2022. The Creativity of Text-to-Image Generation. In *Proceedings of the 25th International Academic Mindtrek Conference (Academic Mindtrek '22)*. Association for Computing Machinery, New York, NY, USA, 192–202. doi:10.1145/3569219.3569352

[53] Jonas Oppenlaender, Rhema Linder, and Johanna Silvennoinen. 2024. Prompting AI Art: An Investigation into the Creative Skill of Prompt Engineering. *International Journal of Human–Computer Interaction* (Nov. 2024). https://www.tandfonline.com/doi/abs/10.1080/10447318.2024.2431761 Publisher: Taylor & Francis.

[54] Comfy Organization. 2024. *ComfyUI*. https://www.comfy.org/

[55] Srishti Palani and Gonzalo Ramos. 2024. Evolving Roles and Workflows of Creative Practitioners in the Age of Generative AI. In *Proceedings of the 16th Conference on Creativity & Cognition (C&C '24)*. Association for Computing Machinery, New York, NY, USA, 170–184. doi:10.1145/3635636.3656190

[56] Xiaohan Peng, Janin Koch, and Wendy E. Mackay. 2024. DesignPrompt: Using Multimodal Interaction for Design Exploration with Generative AI. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference (DIS '24)*. Association for Computing Machinery, New York, NY, USA, 804–818. doi:10.1145/3643834.3661588

[57] D. A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5, 1 (March 1992), 3–14. doi:10.1016/0950-7051(92)90020-G

[58] Donald A. Schön. 2017. *The Reflective Practitioner: How Professionals Think in Action*. Routledge, London. doi:10.4324/9781315237473

[59] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (Dec. 2007), 20–32. doi:10.1145/1323688.1323689

[60] SideFX. 2024. *Houdini - 3D modeling, animation, VFX, look development, lighting and rendering*. https://www.sidefx.com/

[61] Kihoon Son, DaEun Choi, Tae Soo Kim, Young-Ho Kim, and Juho Kim. 2024. GenQuery: Supporting Expressive Visual Search with Generative Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–19. doi:10.1145/3613904.3642847

[62] Erik Stolterman and Jonas Löwgren. 2007. *Thoughtful Interaction Design: A Design Perspective on Information Technology*. MIT Press, Cambridge, MA, USA.

[63] Minhyang (Mia) Suh, Emily Youngblom, Michael Terry, and Carrie J Cai. 2021. AI as Social Glue: Uncovering the Roles of Deep Generative AI during Social Music Composition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–11. doi:10.1145/3411764.3445219

[64] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–26. doi:10.1145/3613904.3642400

[65] Sirui Tao, Ivan Liang, Cindy Peng, Zhiqing Wang, Srishti Palani, and Steven P. Dow. 2025. DesignWeaver: Dimensional Scaffolding for Text-to-Image Product Design. doi:10.1145/3706598.3714211 arXiv:2502.09867 [cs].

[66] Draw Things Team. 2024. *Draw Things: AI-assisted Image Generation*. https://drawthings.ai/

[67] Martin Tricaud and Michel Beaudouin-Lafon. 2024. Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists and designers. In *Proceedings of the 35th Australian Computer-Human Interaction Conference (OzCHI '23)*. Association for Computing Machinery, New York, NY, USA, 175–190. doi:10.1145/3638380.3638395

[68] Priyan Vaithilingam, Elena L. Glassman, Jeevana Priya Inala, and Chenglong Wang. 2024. DynaVis: Dynamically Synthesized UI Widgets for Visualization Editing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 985, 17 pages. doi:10.1145/3613904.3642639

[69] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S. Bernstein, and Ranjay Krishna. 2023. Explanations Can Reduce Overreliance on AI Systems During Decision-Making. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (April 2023), 1–38. doi:10.1145/3579605

[70] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. PromptCharm: Text-to-Image Generation through Multi-modal Prompting and Refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–21. doi:10.1145/3613904.3642803

[71] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. 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)*. Association for Computing Machinery, New York, NY, USA, 1–21. doi:10.1145/3544548.3581388

[72] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 3813–3824. doi:10.1109/ICCV51070.2023.00355 ISSN: 2380-7504.

[73] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. 2019. Pluralistic Image Completion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1438–1447. doi:10.1109/CVPR.2019.00153 ISSN: 2575-7075.

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2242–2251. doi:10.1109/ICCV.2017.244 ISSN: 2380-7504.

A Appendix

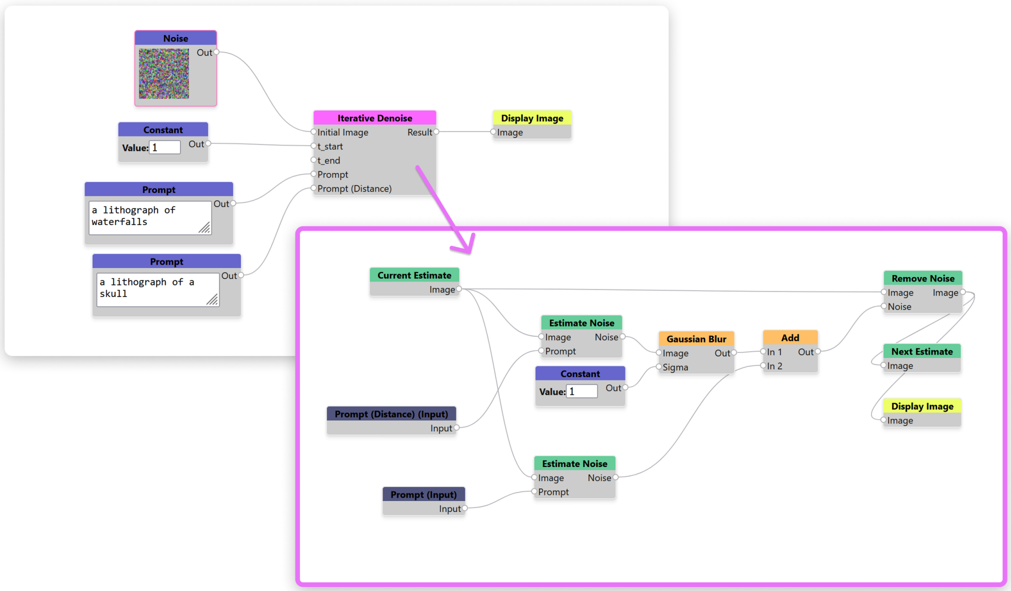

Figure 15: Example of the factorized diffusion workspace as it appears in Noise Pilot. The callout box represents the custom Iterative Denoise node necessary to implement the technique. Two noise estimates are used, each with a different prompt, and one is blurred before being removed to make that prompt visible when the image is viewed at a distance.

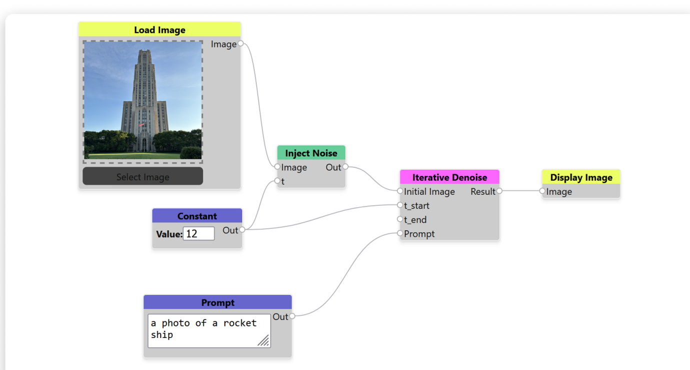

Figure 16: Example of the image to image translation workspace as it appears in Noise Pilot. No custom Iterative Denoise is needed for this technique, noise is simply injected according to the noise schedule, and the constant parameter can determine how close to the original source image the translation results (with lower numbers being less like the original image).

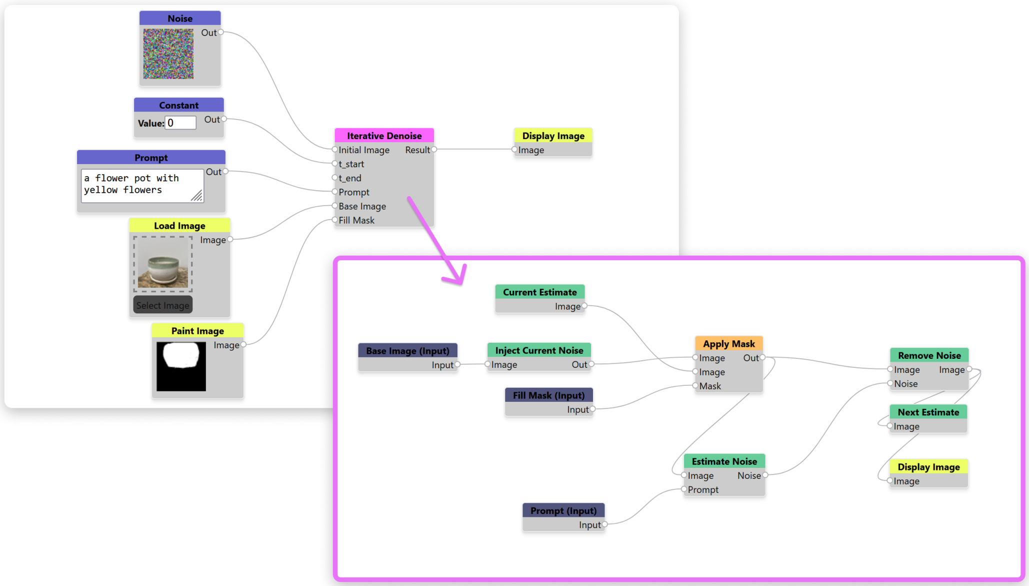

Figure 17: Workspace of the generative fill technique as it appears in Noise Pilot. The callout represents the custom Iterative Denoise node necessary for this technique. Noise is injected into the original image so that when the masked part of the image diffuses it can blend with the surroundings.

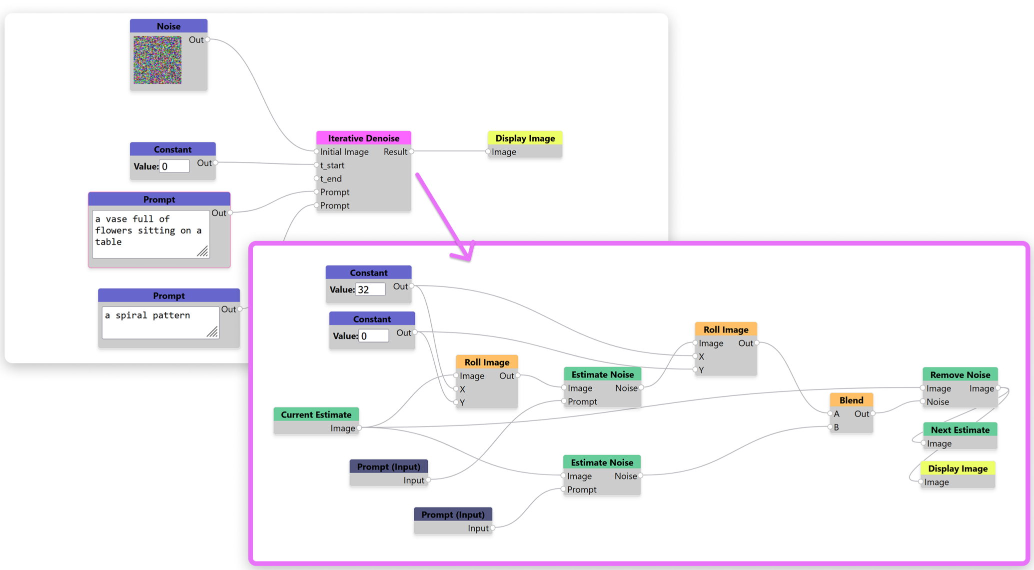

Figure 18: Workspace of the tiling images technique as it appears in Noise Pilot. The callout represents the custom Iterative Denoise node necessary for this technique. Noise is estimated twice, with one of the estimates being rolled equal to half the width of the image to guarantee that the image tiles in the x direction.

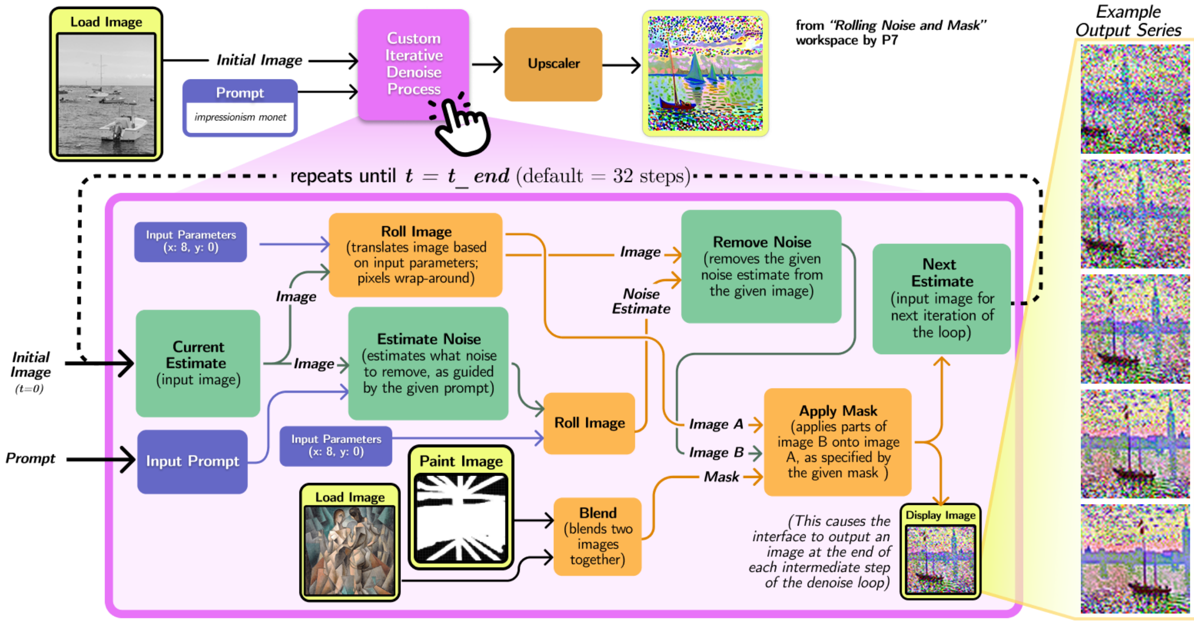

Figure 19: P7 built a custom iterative denoise node into their workspace, "Rolling Noise and Mask". Rather than building a program to achieve a particular concept or vision, P7 progressively added new components to the denoise loop, allowing the discovery of interesting new possibilities to shape their exploration. For example, using "Roll Image" nodes to shift the image along the x-axis caused the system to generate outputs where each intermediate denoised step *also* appeared to horizontally"animate". At right, P7 used a photo of sailboats with art-history inspired prompts to create image series, where boats appear to move in the water as the image diffuses from pure noise.