DeckFlow: A Card Game Interface for Exploring Generative Model Flows

Gregory Croisdale
University of Michigan
Ann Arbor, USA
gregtc@umich.edu

John Joon Young Chung*
SpaceCraft Inc.
Los Angeles, USA
jjyc@spacecraft.inc

Emily Huang
University of Michigan
Ann Arbor, USA
emihuang@umich.edu

Gage Birchmeier
University of Michigan
Ann Arbor, USA
gsbirch@umich.edu

Xu Wang
University of Michigan
Ann Arbor, USA
xwanghci@umich.edu

Anhong Guo
University of Michigan
Ann Arbor, USA
anhong@umich.edu

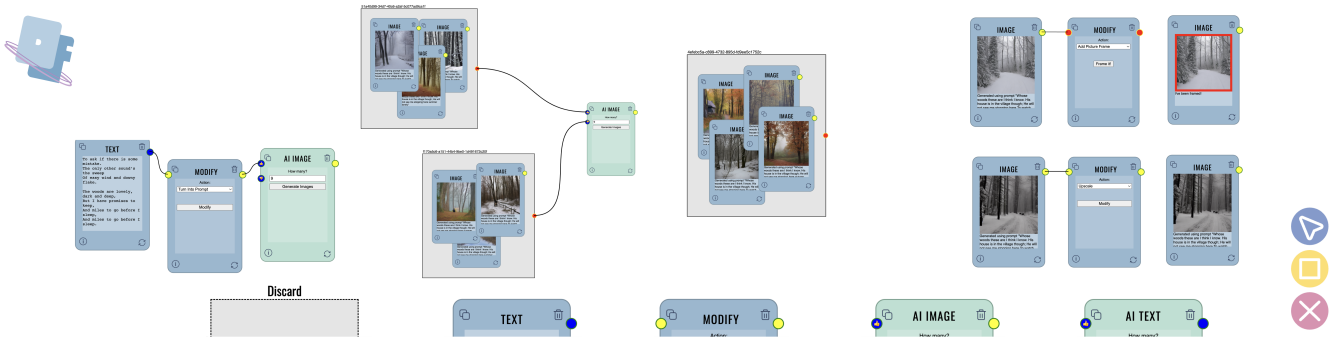

Figure 1: An image generation workflow created in *DeckFlow*.

ABSTRACT

Recent Generative AI models have been shown to be substantially useful in different fields, often bridging modal gaps, such as text-prompted image or human motion generation. However, their accompanying interfaces do not sufficiently support iteration and interaction between models, and due to the computational intensity of generative technology, can be unforgiving to user errors and missteps. We propose *DeckFlow*, a no-code interface for multimodal generative workflows which encourages rapid iteration and experimentation between disparate models. *DeckFlow* emphasizes the persistence of output, the maintenance of generation settings and dependencies, and continual steering through user-defined concept groups. Taking design cues from Card Games and Affinity Diagrams, *DeckFlow* is aimed to lower the barrier for non-experts to explore and interact with generative AI.

CCS CONCEPTS

• **Human-centered computing** → **User interface design**.

KEYWORDS

generative model, multimodal interaction, text-to-image generation

ACM Reference Format:
Gregory Croisdale, John Joon Young Chung, Emily Huang, Gage Birchmeier, Xu Wang, and Anhong Guo. 2023. DeckFlow: A Card Game Interface for Exploring Generative Model Flows. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct), October 29-November 1, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3586182.3615821

1 INTRODUCTION

Recent innovations have drastically increased interest and use in generative AI. One consequence of this newfound broad interest is the frequency of the creation of disparate models with distinct inputs, interface requirements, and individual strengths and weaknesses.

Generative systems such as Automatic1111[3] or LangChain[6] have seen success in encouraging experimentation and comparison within a modality, but they lack the ability to integrate multiple modalities such as image and text into a single workflow, require users to understand and specify complicated parameters to fine-tune the generative process, and do not adequately support iterative exploration and low-fidelity prototyping. Other interfaces like Midjourney[1] or Dall-E 2 [11] obscure these complicated parameters completely to simplify the user experience, but as a consequence, a whole range of model parameters becomes completely inaccessible to the user, potentially diminishing their range of usefulness.

*Work done at the University of Michigan.

UIST '23 Adjunct, October 29-November 1, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0096-5/23/10.
https://doi.org/10.1145/3586182.3615821

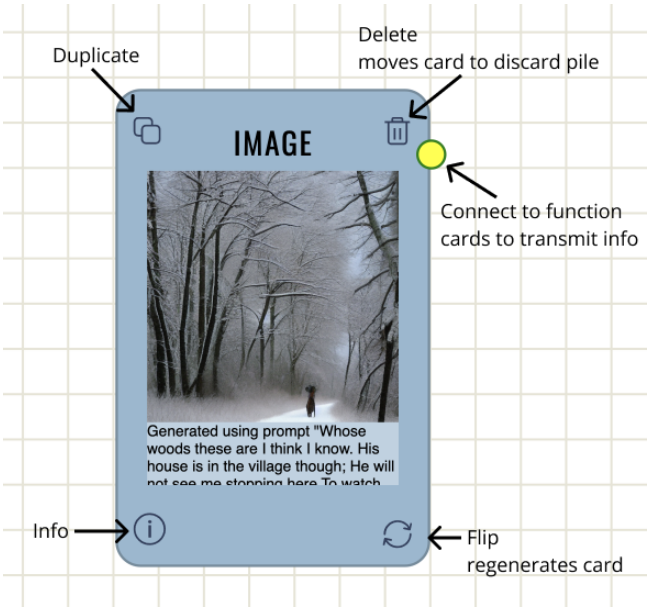

Figure 2: An example of a card and its features in *DeckFlow*.

2 DESIGN GOALS AND INSPIRATIONS

Building upon previous work, we present *DeckFlow* to allow novice users to rapidly and iteratively experiment with flows involving different generative modalities, first instantiated with Text and Images. To accomplish these goals, *DeckFlow* was designed to maximize its approachability, allow the provision of iterative feedback, and give quick and access to modification and recovery.

2.1 Approachability

DeckFlow utilizes card metaphors for most of its system components. Every entity is a part of a card which can be dragged from the user's hand at the bottom of the screen, inspired by card game interfaces like Hearthstone[7] or Inscryption[8]. A user can begin working with *DeckFlow* by first expressing their preference by adding text and images to the screen which are representative of the space they wish to explore.

Once the user has added the initial data, they can group the data according to the concepts they represent. For example, if a user wants to create an image of a red dress, they might add cards into a cluster that represent their desired red color, such as the text "deep velvet rose color" and some reference images, followed by cards in a cluster to represent the style of dress, such as "patterned sundress." Content inside of a cluster is weighted depending on its Y position, similar to how cards in Solitaire are ordered by their strength.

Finally, the user can create new cards by connecting some combination of cards and clusters to a generation card, currently "AI Image" and "AI Text." To connect these entities, the user simply drags an output node near a generation card's input node. The generation cards have a positive input marked with a thumbs up emoji, and negative input marked with a thumbs down emoji. Once the user has indicated their positive and negative feedback, they can click the "generate" button to create more cards.

The complicated parameters involved in generation, such as Stable Diffusion's CFGScale or Sampling Method, are abstracted from the user, but unlike Midjourney[1] or the Dall-E 2 labs interface[11], the system can still utilize its full range of inputs. Rather than being directly manipulated as in Automatic1111[3], however, *DeckFlow* finds ideal parameters through random sampling steered by human feedback.

2.2 Iterative Feedback

When the user requests output from a generation card, *DeckFlow*'s backend generates a random parameter combination, including the model used, from a reasonable distribution. After the user incorporates the newly generated output into clusters, future generations informed by these outputs will sample parameters from a distribution informed by the generation settings of the positive and negative inputs.

Additionally, *DeckFlow* uses the semantic content of the provided input to generate one combined input for the generative model by converting all input into text and using an Large Language Model to combine them, similar to Brade et al.'s Promptify [5]. Unlike the feedback provided to parameter generation, this semantic information can be extracted and utilized, even for user-uploaded media.

2.3 Modification and Recovery

Rather than deletion, *DeckFlow* incorporates a "discard" system in which outputs can be placed into a pile next to the user's hand. This allows users to refer back to previous output which they may have initially thought to be unimportant, but is shown to be useful later.

A card also remembers it lineage, meaning that at any point, the user can replicate the exact workflow which generated an output by simply dragging from its information button, or can regenerate a card with similar settings by clicking its flip button.

2.4 Implementation

DeckFlow is a custom Typescript React App which compiles its visual elements from SVG assets created in Adobe Illustrator.

DeckFlow relies on a C# backend to distribute work to computational nodes or redirect API calls to hosted services and for the maintenance of model output in a database. The backend intelligently breaks workflows into atomic pieces (generally a Python script) and are distributed to nodes by balancing temporal priority and the prevention of model loading and unloading.

3 FUTURE WORK

3.1 Features

3.1.1 Cluster Combination. Currently, each model integrated into *DeckFlow* relies either solely on text or images. When a cluster which contains a different type than is required, a lossy conversion process takes place into the opposing format (using CLIP interrogator to create text from an image or Stable Diffusion to generate an image from text). Text prompts are then combined using GPT-4 with a one-shot example. We are exploring approaches which combine different types of data into a single modality through the use of a shared embedding space.

Previous work [10] enables the isolation of portions of an embedding relevant to a particular feature (for example, a user might indicate that a cluster contains positive examples of a desired color). In conjugation with the clustering mechanism, this would allow further precision in the steering capabilities of the system.

3.1.2 Model Implementation. The implemented models are currently manually written, a practice which is inefficient and potentially unstable. We are developing a Python library to integrate directly with the *DeckFlow* system so that existing models can be trivially be adapted to fit in with the *DeckFlow* system.

3.1.3 Modality Expansion. With the implementation of increasingly capable embedding spaces such as Meta's ImageBind [9], interaction between modalities other than text and images are becoming increasingly simple. Some examples of possible outputs include 3d assets [12], music [2], voice clips [4], or even human motion [13].

REFERENCES

[1] 2023. Midjourney. https://www.midjourney.com/.

[2] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. arXiv:2301.11325 [cs.SD]

[3] AUTOMATIC1111. 2023. *stable-diffusion-webui.*

[4] James Betker. 2023. Better speech synthesis through scaling. arXiv:2305.07243 [cs.SD]

[5] 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. arXiv:2304.09337 [cs.HC]

[6] Harrison Chase. 2022. *LangChain.* https://github.com/hwchase17/langchain

[7] Blizzard Entertainment. 2023. Hearthstone. https://hearthstone.blizzard.com/.

[8] Daniel Mullins Games. 2023. Inscryption. https://www.inscryption.com/.

[9] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 15180–15190.

[10] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. 2017. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning.*

[11] OpenAI. 2023. DALL·E 2. https://openai.com/dall-e-2/.

[12] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. arXiv:2209.14988 [cs.CV]

[13] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. 2022. Human Motion Diffusion Model. arXiv:2209.14916 [cs.CV]