Drawcto: A Multi-Agent Co-Creative AI for Collaborative Non-Representational Art

Manoj Deshpande, Brian Magerko
Georgia Institute of Technology
Atlanta, GA 30308, USA
{manojdeshpande, magerko}@gatech.edu

Abstract

Non-representational art–such as works by Wassily Kandinsky, Joan Mitchell, Willem de Kooning, etc.–showcases diverse artistic expressions and challenges viewers with its interpretive open-endedness and lack of a clear mapping to our everyday reality. Human cognition and perception nonetheless aid us in making sense of, reasoning about, and discussing the perceptual features prevalent in such non-representational art. While there have been various Computational Creative systems capable of generating representational artwork, only a few existing Computational (Co)Creative systems for visual arts can produce non-representational art. How would a co-creative AI that incorporates elements of the human visual perception theory be able to collaborate with a human in co-creating a non-representational art? This paper explores this challenge in detail, describes potential machine learning and non-machine learning approaches for designing an AI agent and introduces a new web-based, multi-agent AI drawing application, called *Drawcto*, capable of co-creating non-representational artwork with human collaborators.

Playing video games can be a highly creative activity, requiring individuals to engage in creative behaviors like content creation, collaborative building, problem solving, etc. (Green and Kaufman 2015; Blanco-Herrera, Gentile, and Rokkum 2019). A model of creativity within AI agents may support new forms of creative gameplay and new applications of AI in game spaces. Inspired by this potential, we focus on exploring a specific creative interaction modality that has its roots in popular sketch-based games like Pictionary or web games like Skribbl.io (mel 2011). Previous research with these games has been limited to the training and development of computationally creative agents (Bhunia et al. 2020; Sarvadevabhatla et al. 2018); our aim is to develop a co-creative system for co-creating non-objective visual art that seeks to invoke the properties of human-computer co-creativity in ways applicable to the study, creation, and play of digital and analog games involving creative aspects.

Computational Creativity (CC) in the visual arts has gained attention since the early days of AARON (Cohen 1995). Over the years, researchers have developed various algorithms and CC systems that are capable of (co)creating artworks in a specific artistic style (Gatys, Ecker, and Bethge 2016), identify and suggest conceptually or visually similar objects (Karimi et al. 2020), produce strokes or pixels to complete a sketch or an image semantically (Su et al. 2020; Iizuka, Simo-Serra, and Ishikawa 2017), etc. While many of these approaches have examined creating representational artwork (e.g., realistic or impressionistic presentations of real-world scenes and objects), little work has been done in exploring how a more abstract, non-representational work could be done by (or in collaboration with) AI–inspired by human cognition and perception–that can discuss its intention behind specific features and composition of the artwork with a human collaborator. In an attempt to address this, we present Drawcto - a web-based, multi-agent AI drawing application capable of co-creating and discussing specific features of non-representational art with a human collaborator.

People often use abstract art and non-representational art interchangeably to refer to the same painting style, yet there are crucial differences between the two terms (Ashmore 1955). Abstract artwork distorts the view of a familiar subject (i.e., a thing, face, body, place, etc.). For example, Picasso distorts a person's face to show different views of the same figure within a single painting. The resulting artwork appears abstracted, but still, there are discernable features and structures intact from the original subject. Figure 1a and 1b show examples of abstract art. Non-representational art, on the other hand, doesn't have a known object or a thing that the artwork is trying to depict. For non-representational art–also known as non-objective art–the artist only uses visual design elements like form, shape, color, line, etc., to express themselves.

Non-representational art represents the spiritual, mystic, non-materialistic, experiential, or creative painting/thought process of the artist (Fingesten 1961), making it challenging to appreciate, contextualize, or understand. For example, Kandinsky's non-objective compositions represent his emotional experience of listening to music, Mondrian's paintings–which only contain straight lines and primary colors–represent "what is absolute, among the relativity of time and space" (Wallis 1960), Pollock's artworks represent the action-painting process, in other words it depicts

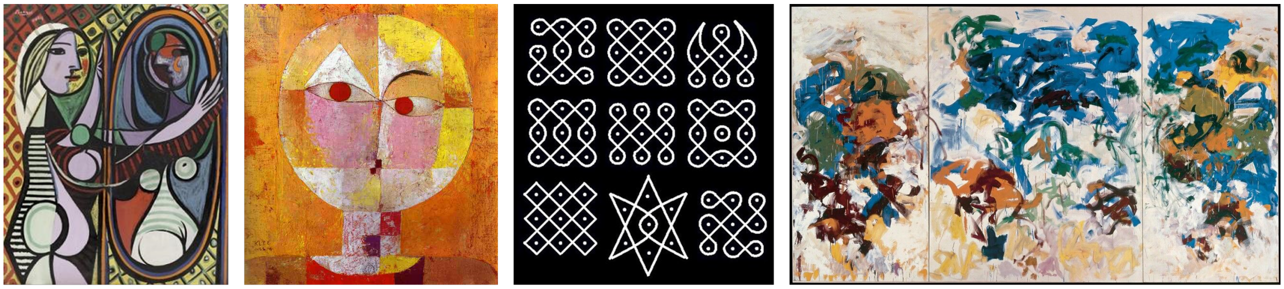

Figure 1: (From left to right)1a:Picasso's painting (Picasso 1932), 1b:Klee's painting (Klee 1922), 1c:Rangoli designs (Balaji 2018), 1d:Joan Mitchell's composition (Bracket 1989)

the forces that lead to its creation. Figure 1c and Figure 1d show examples of non-representational art. Non-representational art generally is not preconceived; instead, it emerges from the artist's in-the-moment interaction with the medium, reflection-in-design process (Schön 1983).

Generating visually sensible content in such a dynamic scenario is the main challenge for developing an AI agent for co-creating non-representational art. We cannot simply train the agent to use object detection or classification to make sense of and generate new strokes as usually there are no recognizable objects. At the same time, we can't generate random strokes as they would not be visually sensible. Therefore, developing an AI that can create various strokes based on its perceptual ability to understand and reason with the quality of strokes made by the human collaborator is the challenge we address in this research.

We utilize the *perceptual organization theory* (or *Gestalt theory*) for the agent(s) to make sense of and generate new strokes while co-creating a non-representational artwork. Gestalt theory describes a finite set of rules that guide and aid the reasoning of our visual system. Some of the gestalt grouping principles are *proximity, balance, continuity, similarity,* etc. (Arnheim 1957). Previously, researchers have used perceptual organization theory for various applications like image segmentation, contour detection, shape parsing, etc. In this paper, we present work that attempts to circumnavigate the "authoring bottleneck" commonly associated with co-creative systems (Csinger, Booth, and Poole 1995) by using perceptual theories (like Gestalt's) to both bootstrap various learning/non-learning approaches to collaborative sketching as well as a basis for affording AI explainability.

We have organized the paper as follows. We examine potential learning and non-learning approaches for developing an AI agent in Related Work. The System Design section describes the current version of Drawcto and explains each component in detail. In the Discussion section, we reflect on the present drawbacks of the three drawing agents. Finally, we share potential future avenues of research we have identified for *Drawcto* in Future Work.

Related Work

In recent years, research on developing image/sketch generation AI has gained a lot of interest. As a result, many research projects and AI architectures have explored image generation from various perspectives and for multiple reasons like co-creating, sketch-based image retrieval, image completion, design ideation, image stylizing, etc. This literature review focuses on diverse learning and non-learning approaches for stroke generation for abstract or non-representational art.

Recurrent Neural Network (RNN)

The Sketch-RNN model (Ha and Eck 2017) is a sequence to sequence variational autoencoder (VAE) that has inspired and informed various co-creative drawing systems. Some examples in recent years are Collabdraw (Fan, Dinculescu, and Ha 2019), DuetDraw (Oh et al. 2018), Suggestive Drawing (Alonso 2017), etc. Sketch-RNN model is trained on QuickDraw dataset (Jongejan et al. 2016) and has learned to express images as short sequential vector strokes. The QuickDraw dataset is a collection of labeled sketches drawn under 20 seconds for a selected object category. Facilitated by quickdraw, the Sketch-RNN model can produce semantically meaningful strokes. Like in Collabdraw, the user and AI collaborate by taking turns to finish a semantically accurate sketch. Also, in projects like DuetDraw and Suggestive Drawing, the capability of the Sketch-RNN model is enhanced by combining it with other features like completing a drawing, transforming an image, doing style transfer, recommending empty-space, etc. RNNs are great for co-creating with line drawing; however, the main challenge while using an RNN is that the training data needs to be sequential vectors, i.e., we can't directly use images to train. We have developed two agents for *Drawcto* using this approach.

Generative Adversarial Network (GAN)

Similar to Sketch-RNN, another influential model is GAN (Goodfellow et al. 2020). In GAN, two neural networks–generator and discriminator–compete with each other to make predictions. The role of the generator is to produce output that highly resembles the actual data and, the role of the discriminator is to identify the artificially created data. Some of the research projects based on GAN are Sketch-GAN (Liu et al. 2019), Doodler GAN (Ge et al. 2021), interactive image to image translation using GAN (Isola et al. 2018), etc. GAN is used in various ways; for example, in Sketch-GAN, it generates strokes for missing parts of the image, while in Doodler GAN, it is used to semantically generate stroke to co-create a surreal creature or a bird. In image-to-image translation (Iizuka, Simo-Serra, and Ishikawa 2017), GAN is used for edge-detection, style generation, etc. Building over GAN, Elgammal et al. proposed CAN (creative adversarial network) (Elgammal et al. 2017) to generate artworks deviating from existing artistic styles resulting in non-representational art with varying degrees of complex textures and compositions. The generative capabilities of GAN are very inspiring; since GAN works with pixels, we can use images to train the network and it can also be very efficient for style transfer.

Transformers

A transformer is a sequence-to-sequence model that uses an attention-mechanism to identify important context, helping it provide better results than an RNN (Vaswani et al. 2017). Based on the Transformer, researchers have developed an open-source ML framework–called BERT (Bi-directional Encoder Representation from Transformer)–which helps computers ‘understand’ the meaning of a word/phrase in the input text by using surrounding text to create context (Devlin et al. 2019). BERT is a pre-trained model which can be fine-tuned using a question and answer dataset; researchers have utilized BERT in various text-to-sketch projects. In CalligraphyGAN (Zhuo, Fan, and Wang 2020), authors combine BERT and conditional GAN to create abstract artworks representing a set of Chinese characters given as input. In Sketch-BERT (Lin et al. 2020), the model learns representations that capture the sketch gestalt. The dual-language image encoder model–CLIP(Contrastive Language-Image Pre-training) (Radford et al. 2021) which uses vision-transformer (Dosovitskiy et al. 2021)–has inspired a whole range of drawing-related projects. For example, in ClipDraw (Frans, Soros, and Witkowski 2021), the agent produces a set of vector strokes in diverse artistic styles satisfying a text input; Fernando et al. combine a dual encoder (Fernando et al. 2021) similar to CLIP with a neural L-system to produce abstract images corresponding to the input text. The dependency on text makes using the transformer a challenge in the context of co-creation; nevertheless, transformer-based models are potent models for image/sketch generation.

Reinforcement Learning

Many research projects deal with learning stroke generation using reinforcement learning (RL). Researchers typically train an agent by letting it interact with a simulated painting environment. The painting environment can be continuous (e.g., SPIRAL (Ganin et al. 2018), Improved-SPIRAL (Mellor et al. 2019), etc.) or differentiable (e.g., StrokeNET (Zheng, Jiang, and Huang 2018), Neural Painter (Nakano 2019), etc.). As a result of learning in this simulated environment, the RL agent learns to produce strokes and abstract artworks. Another approach in training an RL agent is limiting the number of strokes used to represent an object. For example, in Pixelor (Bhunia et al. 2020), the agent is involved in a Pictionary-like game with a human to learn optimal stroke sequence to represent an object. Similarly, Huang et al. use RL to train an agent to paint like humans with only a small number of strokes (Huang, Zhou, and Heng 2019). Interactive learning is another approach for training an RL agent as utilized in Drawing Apprentice (Davis et al. 2015). In Drawing Apprentice, the AI agent analyses the user’s input strokes, recognize drawn objects, and responds with complementary strokes. RL provides various potent methods for training an agent, which we hope to experiment with in the future.

Non-Learning Approaches

There are many non-learning approaches to generate strokes for abstract/non-representational art. For example, AARON (McCorduck 1991) is an intricately authored rule-based AI developed by artist Harold Cohen. Similarly, The Painting Fool (Colton 2012) system emulates a human painter and can describe its artwork through textual description following a set of rules. Drawing Apprentice also has a rule-based AI component to respond to the user’s stroke by tracing, replication, or transformation. Another approach for stroke generation is by using a shape grammar (Stiny 2006). For example, in Broadened Drawspace (Gün 2017), the user engages in a visual-making process with a shape-grammar-based generative system. Stroke-based rendering (SBR) methods also provide many algorithms for stroke generation. SBR algorithms are search or trial-and-error algorithms designed to optimize stroke placement by minimizing an energy function (Hertzmann 2003) or other optimization goals like the number of strokes. Generative capabilities of rule-based systems like AARON inspired us to create a rule-based agent for *Drawcto*.

Summary

The above-related research highlights the following gaps in existing systems for (co)creating abstract artistic images.

All the learning approaches are black-box approaches. The AI agent can’t justify why a certain stroke in a specific location and particular style (color, width, length, weight, etc.) makes sense to the entire composition. Even with rule-based or shape grammar approaches, the agent can’t convey its perception of the composition as a whole. In other words, existing agents can not reason about the visual design or justify their actions while creating the artwork.

Barring Drawing Apprentice, none of the CC systems are capable of co-creating a non-representative art. But, a limitation of Drawing Apprentice is that it does not have any perceptual knowledge bootstraps, so Drawing Apprentice takes a black box learning approach to train the agent which results in it not being able to discuss its intention with a human collaborator. Even with generative systems, only a few research projects focus on creating non-representational art, and almost none discuss the intent behind the composition. It is also interesting to note that the existing systems are either single-agent or multi-feature systems.

The shortcomings mentioned above informed us to develop *Drawcto*- a multi-agent system for co-creating non-representational artwork, which can explain its actions based on the current state of the canvas. In the following section,

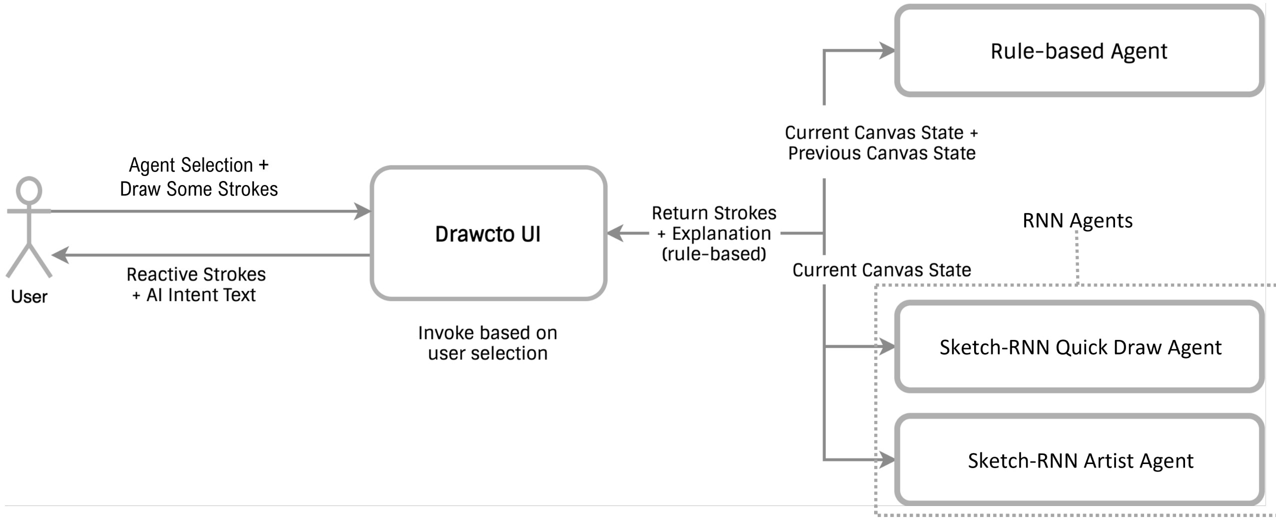

Figure 2: Drawcto system design

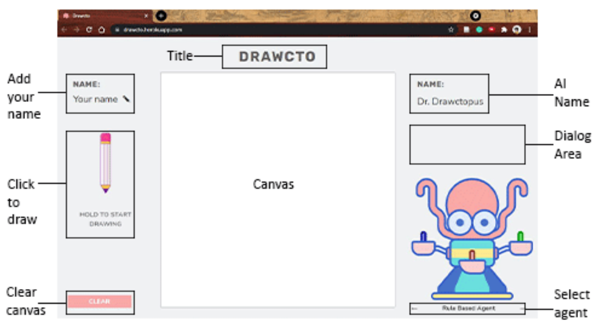

Figure 3: Drawcto user interface

we discuss the system design and various agents' logic for the current prototype of *Drawcto*.

Drawcto System Design

We developed *Drawcto* as an easily accessible web-based application with the graphical interaction happening through a P5.js canvas. The canvas (frontend) communicates to the python server (backend) using HTTP Methods. The backend is responsible for the different AI agents' logic. We developed the python server using the micro web framework Flask. Currently, we are hosting the application on Heroku. As shown in Figure 2, the user draws strokes and selects the agent on the interface; in response, the AI responds with its strokes and a textual description of its stroke intent. We have developed the backend in a modular manner allowing us to add or remove an agent based on its performance. In the following subsections, we describe the user interface and the AI agents in detail.

Drawcto UI

We designed all of the UI features to foster a human-like collaboration between an AI and a human. To anthropomorphize the AI, we named our AI avatar "Dr. Drawctopus" and created a vector image of a cyborg octopus to represent the AI. We chose an octopus to represent our AI with the idea that each tentacle will correspond to a different agent, symbolically conveying that it is part of the same system.

Prior research (Davis et al. 2016) shows that collaboration can be improved by - having permanent screen-presence of the AI character, and dynamically drawing the strokes generated by AI. Hence, we permanently show the AI avatar and its name on the UI, and we animated a visual glyph representing the hand of *Drawcto* moving along the stroke.

Creating non-representational art requires time for reflection-in-design, and turn-taking interaction can facilitate this process. But, turn-taking can be a dynamic process; the most straightforward approach seen in the literature we adopted for Drawcto is simple turn-alternation (Winston and Magerko 2017). However, we needed a way to clearly signal the beginning and end of a turn to the AI with the alternate turn-taking. Therefore, to overcome this, on the interface, we incorporated a pencil that the human collaborator can "pick up" to signal the start of the turn and "place it down" to signal the completion of their turn.

We wanted to present the AI stroke intention to the human collaborator coherently without disturbing the creative collaboration. Hence we decided to show the stroke logic in a dialog box, seeming as if *Drawcto* is communicating. Further, the AI dialogs were written in a way that reflects the friendly persona of the AI.

Along with the UI features mentioned above, the human collaborator can also toggle between the three agents via a clickable arrow. Figure 3 highlights the different parts of the user interface.

Rule-Based Agent

When the rule-based agent is selected, it reacts to the user's stroke(s) based on a set of hand-authored rules. We derived these rules from perceptual grouping theory, such as *bal-*

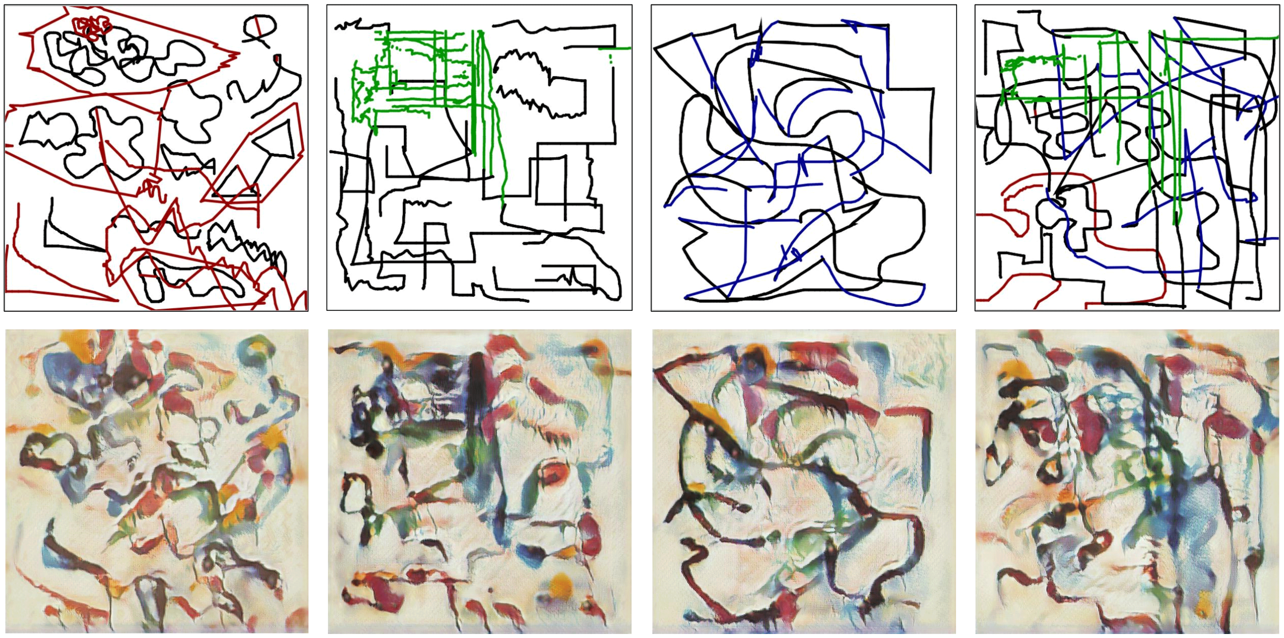

Figure 4: (Top, from left to right) 4a:collaboration with rule-based agent, 4b:collaboration with artist agent, 4c:collaboration with quick draw agent, 4d:collaboration with all agents; (Bottom, from left to right) 4e:stylization of art made with rule-based agent, 4f:stylization of art made with artist agent, 4g:stylization of art made with quick draw agent, 4h:stylization of art made with all agents

ance, symmetry, continuity, closure, etc. (Arnheim 1957). Since non-representational art, in general, is not preconceived, we designed the agent also to behave similarly and not have an end goal across multiple turns. Instead, the agent emulates a painter's reflection-in-design process and reacts only based on the current state of composition on the canvas, primarily based on the collaborator's latest move. Figure 4a shows the result of interaction with the rule-based agent.

The agent makes sense of the strokes strictly based on the observable, salient features on the canvas. We use the OpenCV library to make sense of and extract features from the strokes. The feature set includes - number of contours (for whole canvas or current stroke), the center of mass, white space, and four-way symmetry. We make use of this feature set to traverse a decision tree to find an applicable rule. Some examples of rules include - closing a stroke if the user drew an open stroke; connecting strokes if the user drew more than one separate stroke; enclosing a stroke if the no. of contours is above a threshold; creating similar strokes if the canvas has an empty area, etc.

To better understand how a rule-based agent works consider the following scenario. Assume the human collaborator draws an open shape like a 'U' shape or a polygon with one side missing on the canvas and finishes their turn. Then, a snapshot of the canvas is sent to the backend. With the help of functions in the OpenCV library like *findContour* or *isClosed*, we develop a feature set that indicates that the shape is open. Following this, the agent traverses a predefined decision tree and comes up with two possible moves– to close the open-shape or, to draw a similar but new distorted (scaled up or down, sheared, etc.) open shape. The agent randomly chooses between these moves, produces the relevant stroke on the canvas, and presents a textual description of the rule and why it was triggered to the human collaborator.

RNN Agents

We were curious if we could build a data-driven agent in *Drawcto* that relied on latent information learned from existing artworks or sketch datasets instead of following predefined rules. We developed two separate agents to explore this "authorless" approach in *Drawcto* - the Quick Draw and Artist agents. Both the agents are based on Google's interactive SketchRNN (Ha and Eck 2017) model.

SketchRNN Quick Draw Agent This agent responds by producing a new stroke in exact continuation to the human collaborator's last drawn stroke. The agent utilizes the ml5.js library's SketchRNN model (Nickles, Shiffman, and McClendon 2018), and the main goal for the Quick Draw agent was to explore the stroke generation information learned from the Quick! Draw dataset (Jongejan et al. 2016). However, SketchRNN model requires a particular object category to generate strokes, which is not feasible in non-representational art as there are no objects. We developed two strategies to overcome this- first, we limited the length of the output stroke to have a maximum of 30 points; second, we used the "everything" category in SketchRNN, allowing it to utilize the entire Quick! Draw dataset for stroke generation. These strategies and alternate turn-taking interaction resulted in the Quick Draw agent, based on SketchRNN model that successfully showcased its stroke generation ca-

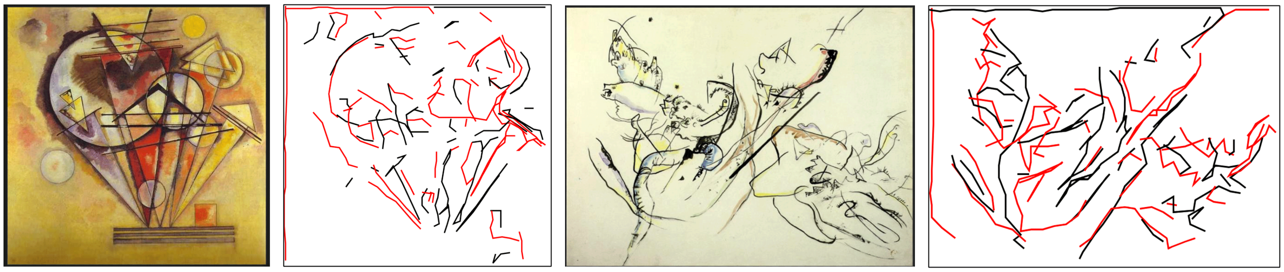

Figure 5: (From left to right) 5a:Kandinsky's painting (Kandinsky 1928), 5b:Edge extraction from the painting, 5c:Kandinsky's painting (Kandinsky 1915), 5d:Edge extraction from the painting

pabilities. Figure 4c shows the result of interaction with the Quick Draw agent.

SketchRNN Artist Agent This agent–to produce strokes–utilizes the SketchRNN model trained on our custom dataset of around 2000 images of non-representational artworks from the web. The goal for the Artist agent was to see if the model would automatically learn visual concepts such as symmetry, shape-completion, balance, etc. However, getting the correct training data was the biggest challenge. To overcome this, we obtained famous non-representational artworks and extracted edges from them and converted them into simple sequential vector drawings. Figure 5 shows examples of the edge extraction we did to collect data. We can see that these images are composed of distinct shapes - like circles, rectangles, etc., have a sense of composition - like positive & negative space, symmetries, etc., and textures are depicted through different densities of the shorter lines. We trained the Artist agent on this data, and Figure 4b shows the interaction with the artist agent.

Discussion

In the related work section, we identified various learning and non-learning approaches that we could take to tackle the challenge of AI generating different semantically accurate strokes while co-creating non-representational art. From that, we chose one non-learning approach - a rule-based Agent, and one learning approach - RNN Agent(s) to incorporate in the current prototype of *Drawcto*. In this section, we reflect on the current limitations of the three co-creative agents.

The rule-based agent currently echoes and acknowledges the user's strokes but rarely produces a novel stroke to the composition. In other words, though the rule-based agent can generate and justify new strokes, it lacks stroke variability and can become predictable after using it a few times.

The RNN agents, on the other hand, especially the quick draw agent, produce a whole variety of strokes due to the diversity and volume of the Quick! Draw data. The dialogues for both the RNN agents give a clue about the training data but fail to reason about a particular stroke. We will have to develop a separate gestalt module to analyze the produced stroke and provide suitable explanations. We notice that a lot of the responses from the artist agent are two lines at right angles. We believe this is because each training image had a boundary. Hence the model learned it as an essential component to any drawing. However, we noticed that the artist agent could respond to the user's stroke, complementing the essence of their drawing style.

Future Work

Adding color, texture, line variation, etc., in a particular artist's style can enhance the co-creative experience. Figure 4e - Figure 4h shows our initial experiment with style transfer; The images in Figure 4 show art in Kandinsky's style. In the future, we hope to develop an agent which will let the user choose to add a stroke in a particular artist's style.

Drawing is an embodied activity, and studies show that maintaining embodied interaction can improve the co-creative drawing experience (Jansen and Sklar 2021). Therefore, we plan to incorporate a robotic arm, which *Drawcto* can use to draw physical strokes on paper or canvas, creating a system where people can draw and co-create physically.

Lastly, For building a learning agent capable of producing stokes based on gestalt theory, Reinforcement learning approaches appear to be a very promising avenue, especially with research like PQA (Qi et al. 2021).

Acknowledgments

We thank Arpit Mathur, Bhavika Devnani, Laney Light, Luowen Qiao, and Tianbai Jia for collaborating to develop this project. We thank Erin Truesdell for providing helpful insights and feedback.

References

Alonso, N. M. 2017. Suggestive Drawing Among Human and Artificial Intelligences. Master's thesis, Harvard University Graduate School of Design.

Arnheim, R. 1957. *Art and visual perception: A psychology of the creative eye*. Univ of California Press.

Ashmore, J. 1955. Some differences between abstract and non-objective painting. *The Journal of Aesthetics and Art Criticism* 13(4):486–495. Publisher: JSTOR.

Balaji, S. 2018. Simple apartment kolam designs.

Bhunia, A. K.; Das, A.; Muhammad, U. R.; Yang, Y.; Hospedales, T. M.; Xiang, T.; Gryaditskaya, Y.; and Song, Y.-Z. 2020. Pixelor: a competitive sketching AI agent. so you think you can sketch? *ACM Transactions on Graphics* 39(6):1–15.

Blanco-Herrera, J. A.; Gentile, D. A.; and Rokkum, J. N. 2019. Video games can increase creativity, but with caveats. *Creativity Research Journal* 31(2):119–131. Publisher: Taylor & Francis.

Bracket, J. M. 1989. Untitled. San Francisco Museum of Modern Art.

Cohen, H. 1995. The further exploits of AARON, painter. *Stanford Humanities Review* 4(2):141–158.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and creativity*. Springer. 3–38.

Csinger, A.; Booth, K. S.; and Poole, D. 1995. AI meets authoring: User models for intelligent multimedia. In *Integration of Natural Language and Vision Processing*. Springer. 283–304.

Davis, N.; Hsiao, C.-P.; Singh, K. Y.; Li, L.; Moningi, S.; and Magerko, B. 2015. Drawing apprentice: An enactive co-creative agent for artistic collaboration. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, 185–186.

Davis, N.; Hsiao, C.-P.; Yashraj Singh, K.; Li, L.; and Magerko, B. 2016. Empirically Studying Participatory Sense-Making in Abstract Drawing with a Co-Creative Cognitive Agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 196–207. Sonoma California USA: ACM.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. arXiv: 1810.04805.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*. arXiv: 2010.11929.

Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: Creative Adversarial Networks, Generating ”Art” by Learning About Styles and Deviating from Style Norms. *arXiv:1706.07068 [cs]*. arXiv: 1706.07068.

Fan, J. E.; Dinculescu, M.; and Ha, D. 2019. collabdraw: An Environment for Collaborative Sketching with an Artificial Agent. In *Proceedings of the 2019 on Creativity and Cognition*, 556–561. San Diego CA USA: ACM.

Fernando, C.; Eslami, S. M. A.; Alayrac, J.-B.; Mirowski, P.; Banarse, D.; and Osindero, S. 2021. Generative Art Using Neural Visual Grammars and Dual Encoders. *arXiv:2105.00162 [cs]*. arXiv: 2105.00162.

Fingesten, P. 1961. Spirituality, mysticism and non-objective art. *Art Journal* 21(1):2–6. Publisher: Taylor & Francis.

Frans, K.; Soros, L. B.; and Witkowski, O. 2021. CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders. *arXiv:2106.14843 [cs]*. arXiv: 2106.14843.

Ganin, Y.; Kulkarni, T.; Babuschkin, I.; Eslami, S. A.; and Vinyals, O. 2018. Synthesizing programs for images using reinforced adversarial learning. In *International Conference on Machine Learning*, 1666–1675. PMLR.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Ge, S.; Goswami, V.; Zitnick, C. L.; and Parikh, D. 2021. Creative Sketch Generation. *arXiv:2011.10039 [cs]*. arXiv: 2011.10039.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM* 63(11):139–144.

Green, G., and Kaufman, J. C. 2015. *Video games and creativity*. Academic Press.

Gün, O. Y. 2017. Computing with Watercolor Shapes. In Çağdaş, G.; Özkar, M.; Gül, L. F.; and Gürer, E., eds., *Computer-Aided Architectural Design. Future Trajectories*, volume 724. Singapore: Springer Singapore. 252–269. Series Title: Communications in Computer and Information Science.

Ha, D., and Eck, D. 2017. A Neural Representation of Sketch Drawings. *arXiv:1704.03477 [cs, stat]*. arXiv: 1704.03477.

Hertzmann, A. 2003. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications* 23(4):70–81.

Huang, Z.; Zhou, S.; and Heng, W. 2019. Learning to Paint With Model-Based Deep Reinforcement Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8708–8717. Seoul, Korea (South): IEEE.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36(4):1–14. Publisher: ACM New York, NY, USA.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2018. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*. arXiv: 1611.07004.

Jansen, C., and Sklar, E. 2021. Exploring Co-creative Drawing Workflows. *Frontiers in robotics and AI* 8:577770.

Jongejan, J.; Rowley, H.; Kawashima, T.; Kim, J.; and Fox-Gieg, N. 2016. Quick, Draw! The Data.

Kandinsky, W. 1915. Untitled, c.1915 - Wassily Kandinsky - WikiArt.org.

Kandinsky, W. 1928. Auf Spitzen (On the points) - Kandinsky, Vassily.

Karimi, P.; Rezwana, J.; Siddiqui, S.; Maher, M. L.; and Dehbozorgi, N. 2020. Creative sketching partner: an analysis of human-AI co-creativity. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 221–230.

Klee, P. 1922. Senecio. Page Version ID: 1028960636.

Lin, H.; Fu, Y.; Jiang, Y.-G.; and Xue, X. 2020. SketchBERT: Learning Sketch Bidirectional Encoder Representation from Transformers by Self-supervised Learning of Sketch Gestalt. *arXiv:2005.09159 [cs, stat]*. arXiv: 2005.09159.

Liu, F.; Deng, X.; Lai, Y.-K.; Liu, Y.-J.; Ma, C.; and Wang, H. 2019. SketchGAN: Joint Sketch Completion and Recognition With Generative Adversarial Network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5823–5832. Long Beach, CA, USA: IEEE.

McCorduck, P. 1991. *Aaron's code: meta-art, artificial intelligence, and the work of Harold Cohen*. Macmillan.

mel. 2011. skribbl - Free Multiplayer Drawing & Guessing Game.

Mellor, J. F. J.; Park, E.; Ganin, Y.; Babuschkin, I.; Kulkarni, T.; Rosenbaum, D.; Ballard, A.; Weber, T.; Vinyals, O.; and Eslami, S. M. A. 2019. Unsupervised Doodling and Painting with Improved SPIRAL. *arXiv:1910.01007 [cs, stat]*. arXiv: 1910.01007.

Nakano, R. 2019. Neural Painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv:1904.08410 [cs, stat]*. arXiv: 1904.08410.

Nickles, E.; Shiffman, D.; and McClendon, B. O. 2018. ml5-library/src/SketchRNN at main · ml5js/ml5-library.

Oh, C.; Song, J.; Choi, J.; Kim, S.; Lee, S.; and Suh, B. 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*, 1–13. Montreal QC Canada: ACM.

Picasso, P. 1932. Girl before a Mirror. Page Version ID: 1023645704.

Qi, Y.; Zhang, K.; Sain, A.; and Song, Y.-Z. 2021. PQA: Perceptual Question Answering. *arXiv:2104.03589 [cs]*. arXiv: 2104.03589.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]*. arXiv: 2103.00020.

Sarvadevabhatla, R. K.; Surya, S.; Mittal, T.; and Babu, R. V. 2018. Game of sketches: Deep recurrent models of pictionary-style word guessing. In *Thirty-second AAAI conference on artificial intelligence*.

Schön, D. A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, Inc.

Stiny, G. 2006. *Shape: talking about seeing and doing*. MIt Press.

Su, G.; Qi, Y.; Pang, K.; Yang, J.; Song, Y.-Z.; and SketchX, C. 2020. SketchHealer: A Graph-to-Sequence Network for Recreating Partial Human Sketches. In *BMVC*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, \.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wallis, M. 1960. The Origin and Foundations of Non-Objective Painting. *The Journal of Aesthetics and Art Criticism* 19(1):61–71. Publisher: JSTOR.

Winston, L., and Magerko, B. 2017. Turn-taking with improvisational co-creative agents. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Zheng, N.; Jiang, Y.; and Huang, D. 2018. Strokenet: A neural painting environment. In *International Conference on Learning Representations*.

Zhuo, J.; Fan, L.; and Wang, H. J. 2020. A Framework and Dataset for Abstract Art Generation via CalligraphyGAN. *arXiv:2012.00744 [cs]*. arXiv: 2012.00744.