SketchAgent: Language-Driven Sequential Sketch Generation

Yael Vinker[1] Tamar Rott Shaham[1] Kristine Zheng[2] Alex Zhao[1] Judith E Fan[2] Antonio Torralba[1]

[1]MIT [2]Stanford University

{yaelvink,tamarott,alexzhao,torralba}@mit.edu {jefan,kxzheng}@stanford.edu

https://sketch-agent.csail.mit.edu/

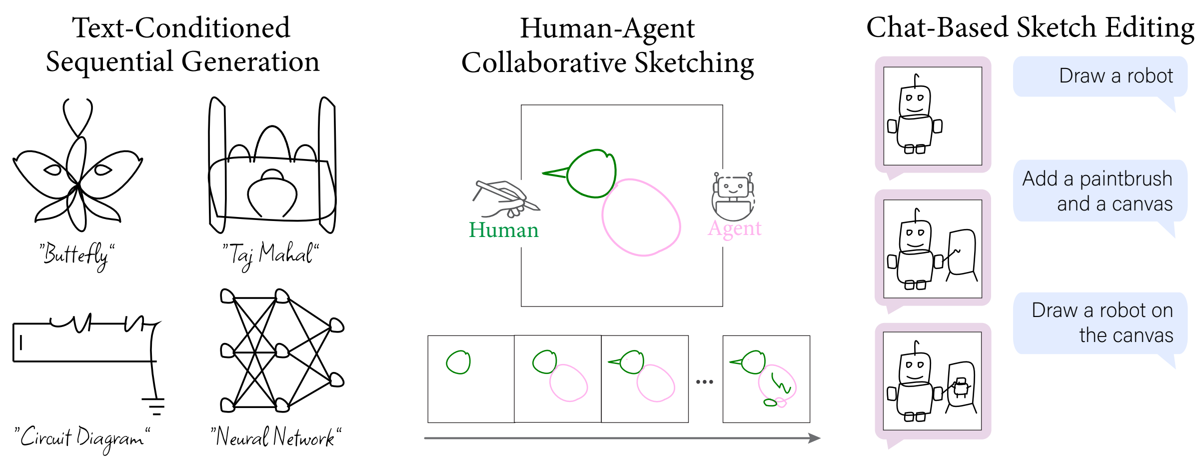


Figure 1. SketchAgent leverages an off-the-shelf multimodal LLM to facilitate language-driven, sequential sketch generation through an intuitive sketching language. It can sketch diverse concepts, engage in interactive sketching with humans, and edit content via chat.

Abstract

Sketching serves as a versatile tool for externalizing ideas, enabling rapid exploration and visual communication that spans various disciplines. While artificial systems have driven substantial advances in content creation and human-computer interaction, capturing the dynamic and abstract nature of human sketching remains challenging. In this work, we introduce SketchAgent, a language-driven, sequential sketch generation method that enables users to create, modify, and refine sketches through dynamic, conversational interactions. Our approach requires no training or fine-tuning. Instead, we leverage the sequential nature and rich prior knowledge of off-the-shelf multimodal large language models (LLMs). We present an intuitive sketching language, introduced to the model through in-context examples, enabling it to "draw" using string-based actions. These are processed into vector graphics and then rendered to create a sketch on a pixel canvas, which can be accessed again for further tasks. By drawing stroke by stroke, our agent captures the evolving, dynamic qualities intrinsic to sketching. We demonstrate that SketchAgent can generate sketches from diverse prompts, engage in dialogue-driven drawing, and collaborate meaningfully with human users.

1. Introduction

Sketching is a powerful tool for distilling ideas into their simplest form. Its fluid and spontaneous nature makes sketching a uniquely versatile tool for visualization, rapid ideation, and communication across cultures, generations, and disciplines [26, 102]. For example, designers use sketches to explore new ideas [39, 103], scientists employ them to formulate problems [48, 72], and children engage in sketching to learn and express themselves [27, 28] (see Fig. 2). Artificial systems, in principle, have the potential to support and enhance human creativity, problem-solving, and visual expression through sketching, adapting flexibly to their exploratory nature [22, 98, 122].

Traditionally, sketch generation methods rely on human-drawn datasets to train generative models [5, 6, 16, 36, 42, 59]. However, fully capturing the diversity of sketches within datasets remains challenging [26], limiting these methods in both scale and diversity. Recent advancements in vision-language models, such as CLIP [78] and text-to-image diffusion [82], have enabled sketch generation methods that reduce reliance on human-drawn datasets [29, 46, 105]. These methods leverage pretrained model guidance and differentiable rendering [58] to optimize parametric curves, creating sketches that go beyond predefined styles and categories.

While representing a significant step toward a general-purpose sketching system, these methods lack a crucial aspect of human drawing: the *process* itself. Current methods, though versatile, optimize all strokes simultaneously, making the intermediate sketching steps meaningless. As a result, the sketch cannot be decomposed into a coherent sequence of strokes that reflects the drawing process. In contrast, humans draw iteratively, stroke by stroke, incorporating visual feedback and continuously adapting—a dynamic, evolving process that fosters creativity, ideation, and communication [52, 88, 101].

In this work, we introduce SketchAgent, a sketch generation agent that leverages the prior knowledge and sequential nature of multimodal large language models (LLMs) to enable versatile, progressive, language-driven sketching. Our agent can generate sketches across a wide range of textual concepts—from animals to engineering principles (Fig. 1, left). Its sequential nature facilitates interactive human-agent sketching and supports iterative refinement and editing through a chat-based dialogue (Fig. 1, right).

Unlike vision-language models that directly generate images from text [75, 80, 82], multimodal LLMs [1, 2, 15, 56, 64, 74, 97] accept text and images as input but only output text. To produce visuals, they either utilize external "tools" (such as calling a text-to-image model) or are prompted to generate executable code (e.g., Python [43], SVG [9]) to create charts, diagrams, or graphics. However, prompting for such representations to directly produce sketches often results in a mechanical appearance with uniform, precise shapes that lack the subtle irregularities and spontaneous qualities characteristic of human sketches (see Fig. 3B). Additionally, despite their robustness in textual tasks, these models often struggle with fine-grained spatial reasoning [41, 118] as they are primarily optimized for text, making sketch editing more challenging.

To address these limitations, we introduce an intuitive sketching language that enables an off-the-shelf multimodal LLM agent to "draw" sketches on a canvas by providing string-based actions, without additional training or fine-tuning. We define the canvas as a numbered grid, allowing the agent to reference specific coordinates (e.g., `x2y8`) to enhance its spatial reasoning capabilities. We represent a sketch as a sequence of semantically meaningful strokes, each defined by a series of such coordinates. We leverage In-Context Learning (ICL) [7, 51] to introduce the agent to the new representation, and Chain of Thought (CoT) [108] to enhance its planning capabilities. Given a sketching task, the agent produces a textual response following our representation, which we process by fitting a smooth Bézier curve to each coordinate sequence. The curves are then rendered onto the canvas to form the final sketch. We find this approach useful in emulating a more natural sketch appearance. For collaborative sketching, the canvas remains accessible to both the user and the agent throughout the session. The agent generates strokes sequentially and pauses according to an adjustable stopping token, allowing the user to add their own strokes directly to the canvas. These strokes are then integrated into the agent's sequence, enabling it to continue drawing, with real-time canvas updates.

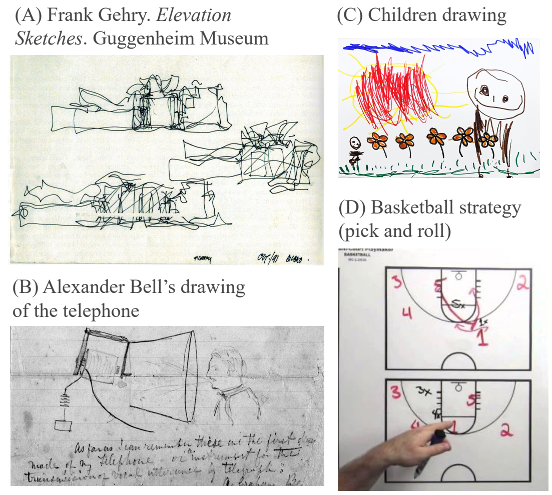


Figure 2. Examples of sketches used across disciplines and goals. (A) Ideation and design: *Process Elevation Sketches* by the architect Frank Gehry, Guggenheim Museum. (B) Engineering: Alexander Bell's telephone drawing. (C) Expressing emotions: Children's sketches. (D) Visual communication: Planning and communicating game strategy in basketball.

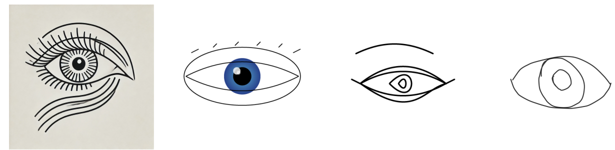
We demonstrate SketchAgent's capability to generate sketches of diverse concepts while capturing the inherently sequential and dynamic nature of sketching. We showcase our agent's ability to collaborate effectively with humans in real time to create novel and meaningful sketches. Our method is the first to leverage pretrained multimodal LLMs for sequential sketching without additional training, paving the way for a general-purpose artificial sketching system that supports iterative, evolving interactivity.

2. Related Work

Sketch Generation Early methods approached sketch generation by designing image filters to simulate sketch-like effects [10, 109]. With the advent of deep learning, data-driven approaches emerged to address a range of sketch-related tasks [117], including category-conditioned sketching [42, 76, 93], object sketching [59, 62], scene-sketching [12, 57, 60, 114], sketch completion [6, 63, 94], portrait drawing [3, 120, 121], part-based generation [6, 37, 42, 129], and more. While sketch data collection has been broadly explored [21, 34, 40, 71, 85, 113], the wide variation in sketch styles and their adaptation to specific tasks [23] makes collecting datasets that encompass this diversity challenging. For example, QuickDraw [47], the largest available sketch dataset with 50 million sketches, covers only 345 object categories and primarily focuses on simple, iconic representations. This limits data-driven methods to the style, abstraction level, and concepts seen during training. Recently, large pretrained vision-language models [75, 78, 80, 82, 84] have shown remarkable text-to-image generation capabilities by leveraging extensive visual knowledge from billions of training images [89]. While these models can be prompted to generate sketch-like images (see Fig. 3A), they do so in a single step and in pixel space, lacking the sequential, stroke-based process of human sketching. Subsequent approaches [14, 29, 31, 46, 105, 106, 115, 116, 124] leverage the priors of these models to guide an iterative optimization of parametric curves, with a differentiable rasterizer [58] linking pixel and vector representations. While producing vector sketches, the final strokes lack order and semantic meaning, and the optimization-based approach overlook the sequential aspect of the sketching *process*, making these methods suboptimal for collaborative sketching.

Sequential and Collaborative Sketching Collaborative human-machine sketching holds promise in enhancing creativity, ideation, communication, and learning, as explored in various fields, including human-computer interaction (HCI) [17, 45, 49, 50, 53, 54], computer graphics [55, 96], robotics [86, 87], cognitive science [24, 25, 35, 67], learning sciences [18, 38, 104], and more. Central to collaborative sketching is its sequential, adaptive, and dynamic process, with each action carrying intent. Existing methods employ diverse training strategies to account for the discrete nature of sequential sketches, including reinforcement and adversarial learning [32, 68, 129], multi-agent referential games [69, 77], transformers [5, 6, 11, 33, 61, 81, 112], and more. SketchRNN [42] is a pioneering work in this area, introducing the QuickDraw dataset [47], a crowd-sourced collection of real-time sketch sequences made by users. They utilize this dataset to train a recurrent neural network for sequential sketch generation, which was later shown [24, 73] to have potential for human-machine collaboration. However, this approach remains constrained by the predefined categories encountered during training.

Multimodel LLMs for Content Creation LLMs [7, 19, 79, 100] and multimodal LLMs [1, 2, 15, 56, 64, 74, 97] receive text as input (or text and images for multimodal) and output text. To enable visual content generation, these models are often paired with external "tools" that extend their functionality [44, 90, 111, 119]. For example, ChatGPT [74] generates images by internally calling a separate model, DALLE-3 [4]. Another approach involves prompting models to produce code in languages like Python [43], Processing [92], SVG [9], or TikZ [8] that can be rendered into visuals such as graphs, charts, and vector graphics. However, such code-generated content often looks rigid, with uniform and overly precise shapes that lack the subtle irregularities and spontaneous qualities characteristic of freehand sketches (see Fig. 3B). In contrast, we propose a sketching language grounded in spatial information that encourages the model to produce a more natural sketch appearance, which we then process into vector graphics. Common strategies for enhancing LLMs capabilities include Chain-of-Thought prompting [13, 70, 83, 91, 127], which breaks down tasks into smaller, logical steps to mimic human reasoning, and In-Context Learning (ICL) [7, 20, 95, 123, 125], where examples of input-output pairs are provided to help the model infer task patterns.



(A) DALLE3 [4] (B) LLMs [2] (SVG) (C) SketchAgent (D) Human [47]

Figure 3. Sketch appearance. (A) Text-to-image diffusion models operate in pixel space, lacking the sequential nature of sketches. (B) Prompting LLMs to produce visuals with SVG results in a uniform, mechanical appearance. (C) Sketches produced by our agent appear less mechanical, more closely resembling the nature of (D) Human sketches, which are often spontaneous and irregular.

3. Preliminaries

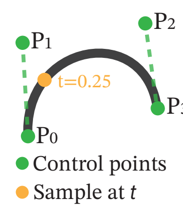


Figure 4. Cubic Bézier curve.

Vector Graphics and Bézier Curves Vector graphics allow us to create visual images directly from geometric shapes such as points, lines, curves, and polygons. Unlike raster images (represented with pixels), vector graphics are resolution-free, more compact, and editable. SVG [110] is an XML-based format for storing vector graphics, popular for its scalability and compatibility with modern web browsers. The process of transferring vector graphics into pixel images is called rasterization or rendering. Cubic Bézier curves are commonly used to represent sketches in vector graphics. A cubic Bézier curve (Fig. 4) is a smooth parametric curve defined by four points: a start point P_0, an end point P_3, and two control points P_1 and P_2 that shape the curvature . The set $P = \{P_0, P_1, P_2, P_3\}$ is often referred to as the curve's control points. The curve is described by the following polynomial equation:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3, \quad (1)$$

where $t \in [0, 1]$ is a parameter that moves the point along the curve from P_0 at $t = 0$ to P_3 at $t = 1$.

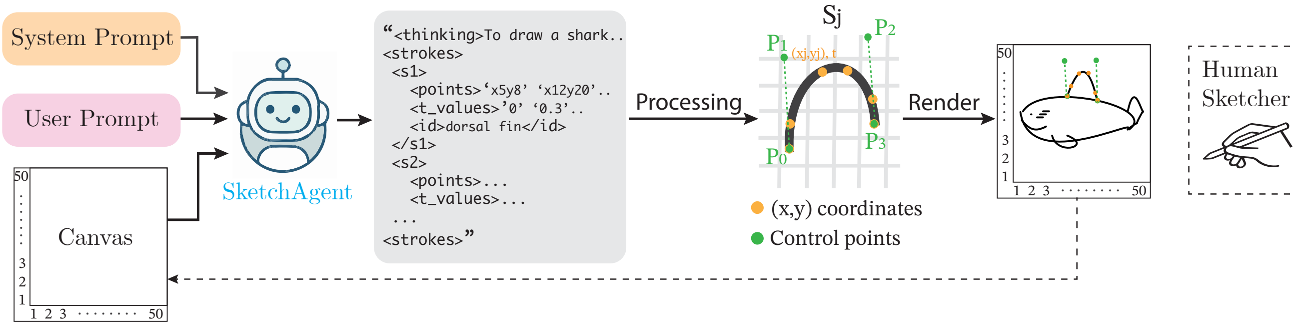


Figure 5. Method Overview. SketchAgent (blue) receives drawing instructions and generates a string representing the intended sketch. Inputs include: (1) a system prompt (orange) introducing the sketching language and canvas, (2) a user prompt (pink) specifying the task (e.g., “draw a shark”), and (3) a numbered canvas. The agent’s response outlines a sketching strategy (in thinking tags) and a sequence of strokes defined by coordinates, which are processed into Bézier curves and rendered onto the canvas.

4. Method

Our goal is to enable an off-the-shelf pretrained multimodal LLM to draw sketches based on natural language instructions. An overview of our pipeline is illustrated in Fig. 5. We utilize a frozen multimodal LLM (“SketchAgent” shown in blue), which receives three inputs: (1) a system prompt containing guidelines for using our new sketching language, (2) a user prompt with additional task-specific instructions (e.g., “*Draw a shark*”), and (3) a blank canvas on which the agent can draw. Based on the given task, the agent generates a textual response, representing the sequence of strokes to be drawn, which we then process into vector graphics and render onto the canvas. The canvas can then be reused in two ways: it can be fed back into the model with an updated user prompt for additional tasks and editing, or it can be accessed by a human user who can draw directly on it to facilitate collaborative sketching. Next, we describe each component of the pipeline.

The Canvas Although multimodal LLMs demonstrate remarkable reasoning abilities, they often struggle with spatial reasoning tasks [30, 66, 99]. We present a simple example (see Fig. 6) to illustrate how this limitation affects the naive use of these models for sketch generation and interactive sketching. We provide GPT-4o [74] with an image depicting a simple line drawing of a partial house featuring five numbered points (from 1 to 5), and ask it to identify which points should be connected to complete the house. While the model correctly identifies the pair of points, it fails to select the correct pixel coordinates when given a basic `draw_line` tool that connects two points, even after multiple attempts. To enhance the model’s spatial reasoning ability, we utilize a numbered canvas that forms a grid. This grid features numbers (1 to 50) along the x-axis and the y-axis (Fig. 5, left). Each cell is uniquely identified by a combination of the corresponding x-axis and y-axis numbers (e.g., the bottom-left cell is `x1y1`). The agent interacts with the canvas by specifying desired `(x,y)` coordinates.

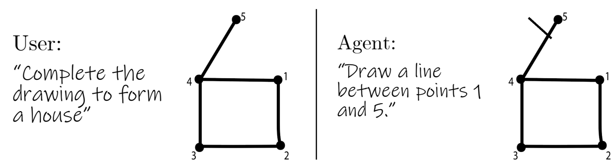
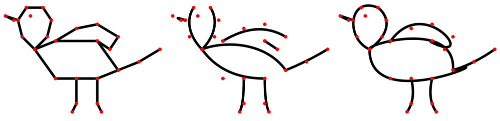


Figure 6. Although excelling in visual reasoning, multimodal LLMs often struggle to translate these abilities into spatial actions. In this example, GPT-4o [74] intends to draw a line between points 1 and 5 but fails to execute this with a `draw_line` function that accepts pixel coordinates.

Sketch Representation We define a sketch as a sequence of n ordered strokes $S = \{S_1, S_2, ...S_n\}$. Each stroke S_i is defined by a sequence of m cell coordinates on the grid: $S_i = \{(x_j, y_j)\}_{j=1}^m$, represented in string format as: `<points>x1y1, x15y20, ...</points>`.

A naive approach to processing the textual sequence of coordinates would be to use a polyline, connecting consecutive points with line segments. However, our grid-based representation sparsifies the canvas, resulting in a non-smooth and unnatural appearance when using polylines (see Fig. 7, left). To achieve a smoother appearance, an alternative approach is to treat the coordinates as a sequence of control points defining smooth curves. However, as illustrated in Fig. 4, the control points often do not lie directly on the curve. Consequently, if the agent aims for a stroke that passes through specific coordinates, it must derive the control points that define this stroke, which is challenging.

We propose an alternative approach: we treat the specified `(x,y)` coordinates as a set of desired points sampled **along** the curve, and fit a smooth Bézier curve to them (Fig. 7, right). To accommodate curves with complex curvature, we also task the model with determining **when** each point on the curve should be passed through, corresponding to the t value described in Eq. (1). Thus, for each stroke S_i, the agent provides a set of m sampled points $S_i = \{(x_j, y_j)\}_{j=1}^m$, along with a corresponding set of t values: $T_i = \{t_j\}_{j=1}^m$. Based on these, we fit a cubic Bézier



(A) Polyline (B) Control Points (C) Ours

Figure 7. Methods for processing the agent's coordinate sequence (in red): (A) Polyline results in an unnatural appearance. (B) Directly using coordinates as Bézier control points is challenging as they do not lie on the curve. (C) Fitting a Bézier curve to sampled coordinates provides smoother results.

curve to the sampled points by solving a system of linear equations using least squares, where the unknowns are the control points $P = \{P_0, P_1, P_2, P_3\}$:

$$P = \text{argmin}_P ||AP - B||, \quad (2)$$

where $A \in \mathbb{R}^{m \times 4}$ contains the cubic Bézier basis functions evaluated at specific t_j values (as described in Eq. (1)), and $B \in \mathbb{R}^{m \times 2}$ contains the m sampled points $\{(x_j, y_j)\}_{j=1}^m$. The least squares solution minimizes the error between the fitted Bézier curve and the sampled points. For long sequences resulting in a large fitting error, we recursively split the curve. Additionally, we account for Bézier curves of lower degrees, including quadratic curves, linear lines, and points. Upon completing this process, we render the parametric curves onto the canvas.

Drawing Instructions We provide the model with a system prompt and a user prompt (marked in orange and pink in Fig. 5). In the system prompt, we supply the agent with context about its expertise ("*You are an expert artist specializing in drawing sketches*") and introduce it to the grid canvas along with examples of how to use our sketching language for drawing single-stroke primitives (full prompts are provided in the Appendix). The system prompt is fixed and can be applied to a variety of sketching tasks. The user prompt includes a description of the desired task and an example of a simple sketch of a house drawn with our sketching language. We find this to be crucial in assisting the agent with preserving the correct format that could be parsed directly [7]. The agent is tasked with responding in the format shown in the gray text box in Fig. 5. In the <thinking> tags, the agent is tasked to outline the overall sketching strategy [108]. This typically includes describing the different components of the sketch, the intended sketching order, and the overall placement of each part. The agent is also tasked with providing an ID tag following each stroke, which is useful for further analysis and for producing annotated sketches in scale.

4.1. In-Chat Editing and Collaborative Sketching

The above process can be repeated iteratively to support multiple sketching tasks and interactions. Text-based sketch editing in a chat dialogue is enabled by feeding the rendered

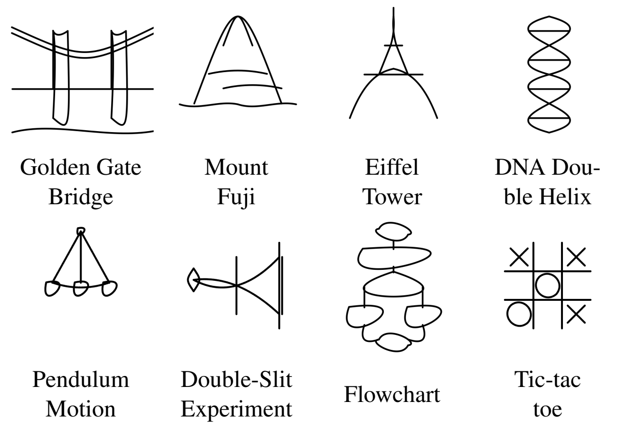


Figure 8. Sketches produced by SketchAgent for concepts beyond pre-defined categories. The textual input describing the desired concept shown below each image.

canvas back to the agent (see dashed arrow in Fig. 5) and updating the user prompt with the desired edits. To support collaborative human-agent sketching, the canvas remains accessible to both the human user and the agent throughout the entire sketching session. We define an adjustable stopping token, </s{j}>, which instructs the agent to pause generating the sequence at stroke number j. We then process and render the generated strokes onto the canvas up to that point, then the user can add strokes directly to the canvas to continue the sketch. The user-drawn strokes are processed and converted into the agent's format by reversing our fitting process, i.e., sampling each stroke at multiple t values (as shown in Eq. (1)), and selecting the points closest to each cell's center on the grid. The converted user strokes are then chained with the agent's sequence, after which the agent resumes sketching until the next stopping token.

5. Results

We evaluate the performance of our method qualitatively and quantitatively across a selected set of sketching tasks. Additional tasks, evaluations, and examples are provided in the Appendix. All results presented in the paper were generated using Claude3.5-Sonnet [2] as our backbone model, unless stated otherwise.

5.1. Text-Conditioned Sketch Generation

Figures 1 and 8 demonstrate SketchAgent's capability to generate sketches of various concepts that extend beyond standard categories, which includes scientific concepts (e.g., "the double-slit experiment", "pendulum motion"), diagrams (e.g.,"circuit diagram", "a flowchart"), and notable landmarks (e.g., "Taj Mahal", "Eiffel Tower"). More examples are provided in the Appendix. To quantitatively evaluate text-conditioned generation we utilize the Quick-Draw dataset [47]. We randomly sample 50 categories (out of 345), and apply our method to generate 10 sketch in-

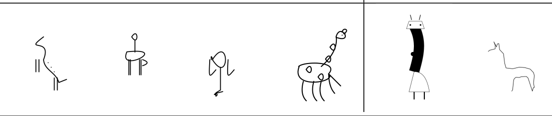
	GPT-4o	GPT-4o -mini	Claude3 Opus	Claude3.5 -Sonnet*	Claude3.5 -Sonnet (SVG)	Human (QD [47])
Top1	0.15 ±0.04	0.04 ±0.03	0.13 ±0.04	0.23 ±0.05	0.23 ±0.04	0.27 ±0.07
Top5	0.30 ±0.06	0.10 ±0.04	0.27 ±0.05	0.44 ±0.03	0.43 ±0.06	0.49 ±0.06
Vis.						

Table 1. Sketch recognition evaluation. Average Top-1 and Top-5 sketch recognition accuracy computed with CLIP zero-shot classifier on 500 sketches from 50 categories. The last row visualizes one sample from each experiment. *Indicates our default settings, which receives the highest accuracy among all models.

stances per category, resulting in 500 sketches in total. Following common practice [105, 106, 115, 116], we utilize a CLIP zero-shot classifier [78] to evaluate how well the generated sketches depict the intended category. We compare the performance of different multimodal LLMs by repeating the same process with GPT-4o-mini [74], GPT-4o [74], and Claude3-Opus [2] as our backbone model (in addition to Claude3.5-Sonnet [2], our default backbone). As a baseline, we include human-drawn sketches sampled from the QuickDraw dataset [47]. The average Top-1 and Top-5 sketch classification accuracy are presented in Table 1. As can be seen, human sketches achieve the highest recognition accuracy, with Claude3.5-Sonnet performing best among all models, approaching human-level rates under the CLIP-score metric. More evaluation of confusion patterns and visualization of the data are provided in the Appendix.

We additionally compare to prompting Claude3.5-Sonnet to directly generate SVGs using the following prompt: *"Write SVG string that draws a sketch of a <concept>. Use only black and white colors"*. The corresponding scores are shown in the fifth column of Tab. 1. While this approach achieves recognition scores comparable to those of SketchAgent, the outputs are often characterized by uniform and precise shapes, failing to replicate the fluidity and natural irregularity of free-hand human sketches (e.g., Fig. 3). To evaluate how "human-like" our agent's sketches appear, we conduct a two alternative forced choice (2AFC) user study with 150 participants. Each participant was presented with pairs of sketches depicting the same object class produced by different methods, and asked to choose the sketch they believed was human-drawn. 150 sketches across 50 object classes were tested, comparing three methods: direct prompting, SketchAgent, and human sketches from QuickDraw (see Appendix for details). Results indicate SketchAgent's drawings appeared more human-like, being chosen as human-drawn in $74.90 \pm 3.35\%$ of cases when compared with direct prompting. When compared to human drawings, users slightly preferred human sketches ($54.68 \pm 4.61\%$) over SketchAgent's, while direct prompting was chosen only $38.9 \pm 5.55\%$ of the time.

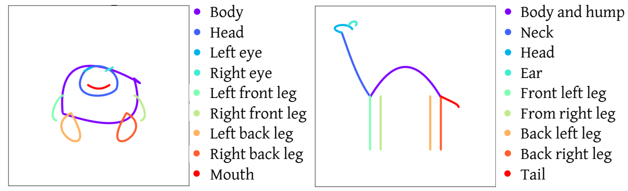


Figure 9. SketchAgent gradually draws stroke-by-stroke, each stroke is annotated by the agent with a semantic meaning.

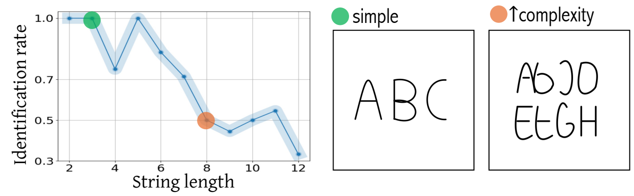


Figure 10. Recognition rate as a function of sketch complexity.

Lastly, to quantitatively analyze the effect of concept complexity, we study the case of drawing letters. We systematically increase sketch complexity by adding letters to the target concept (e.g., from 'ABC' to 'ABCDEFGH') and count the number of correctly recognized letters in each sketch. The graph in Fig. 10 shows that performance decreases as the complexity increases.

5.2. Sequential Sketching

Figure 9 shows stroke-by-stroke sketch generation by SketchAgent, with the labels on the right indicating the sketching order and the meaning our agent associates with each generated stroke (see Appendix for more examples). Stroke annotation during generation is enabled by utilizing the prior of the backbone LLM, providing a valuable feature for analysis and data collection [37, 65, 107, 126, 128]. In Fig. 11, we illustrate why accounting for the sequential nature of sketching more closely emulates the process of human drawing. We present the sketch creation process of SketchAgent alongside SVGDreamer [116], SketchRNN [42], and a human sketch sampled from QuickDraw [47]. SVGDreamer (first row), is an optimization-based method, where a set of randomly initialized parametric curves (leftmost column) are iteratively refined to form a sketch, guided by a pretrained text-to-image diffusion model [82]. This process is time-consuming, taking 2000 iterations (1.6 hours), which makes it unsuitable for interactive sketching. While the final sketch (rightmost column) appears detailed and artistic due to the powerful vision backbone, the intermediate sketching and individual strokes lack clear semantic meaning. In contrast, SketchRNN (second row) is a sequential generative model trained on human-drawn dataset, producing sketches in real-time with strokes added progressively, emulating closer a human-like sketching pro-

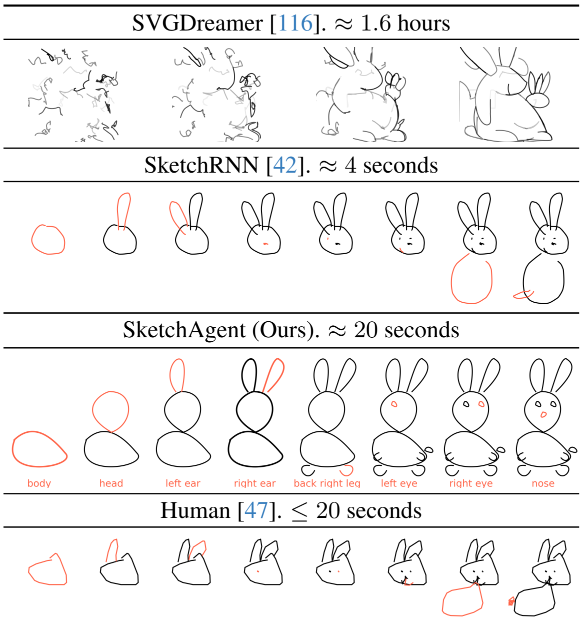


Figure 11. Sequential sketching process. SVGDreamer [116] requires 2000 iterations (1.6 hours) with intermediate steps lacking semantic meaning. SketchRNN [42] operates in real-time with coherent steps but is limited to QuickDraw categories. SketchAgent draw gradually with meaningful strokes and no category restrictions. Human sketches evolve through gradual, meaningful steps.

cess (as shown in the last row). Similarly, SketchAgent (third row) produces sketches gradually, with each stroke carrying a semantic meaning, by utilizing the sequential nature of its backbone model. While SketchRNN is restricted to generating sketches only within the 345 categories it was trained on, SketchAgent leverages the extensive prior knowledge of its backbone multimodal LLM, enabling it to create sketches of general visual concepts.

We use the set of 500 samples described in Sec. 5.1 to quantitatively analyze the sequential nature of our agent's sketches compared to human drawings. On the left of Fig. 12, we present histograms comparing the number of strokes in QuickDraw sketches (orange) and our sketches (blue). Most QuickDraw sketches contain 1 to 6 strokes, while our sketches show a broader distribution, peaking between 5 to 10 strokes. This suggests that, on average, QuickDraw sketches appear more abstract. To ensure a balanced comparison of sketches with similar levels of abstraction, we select sketches from both groups with a similar number of strokes (the largest intersection is found in sketches with 4-7 strokes, comprising 204 of our sketches and 120 from QuickDraw) and measure the change in CLIPScore as a function of the accumulated number of strokes (Fig. 12, right). Both QuickDraw and our sketches exhibit a generally similar pattern, with CLIPScore increasing as more strokes are added, suggesting that sketches become progressively more recognizable as they evolve.

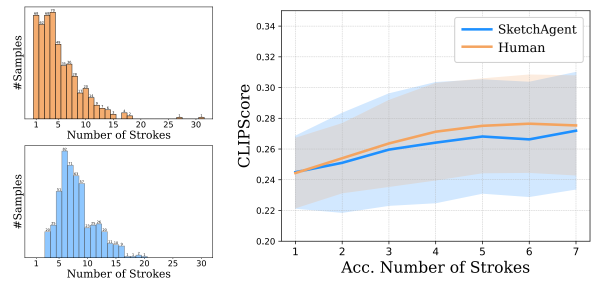


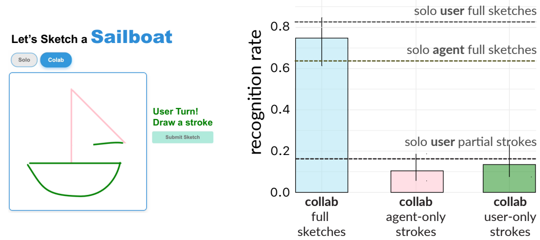
Figure 12. Sequential sketching analysis of SketchAgent (blue) and Humans [47] (orange). Left: Histograms of stroke distribution per sketch, showing QuickDraw sketches are more abstract on average. Right: CLIPScore as a function of the accumulated number of strokes for sketches containing 4-7 strokes, showing a similar recognition pattern over time.

5.3. Human-Agent Collaborative Sketching

We demonstrate the potential of our system for facilitating interactive human-agent collaboration, resulting in semantically meaningful and recognizable sketches. We design a web-based collaborative sketching environment (Fig. 13A) where users and SketchAgent take turns drawing on a shared canvas to create a recognizable sketch from a given textual concept. Following the evaluation protocol in collabdraw [24], we select 8 simple concepts, based on the agent's demonstrated ability to draw them independently, to focus evaluations on assessing the impact of *collaboration*. Participants sketched concepts in two modes: *solo*, where users drew independently, and *collab*, where users and SketchAgent collaborated, adding one stroke at a time until either was satisfied with the drawing. We collect sketches from 30 participants, resulting in 480 sketches in total. Average CLIP recognition rates are shown in Figure 13B. Collaboratively produced sketches (blue) achieve recognition levels close to those made solely by users and higher than those produced by the agent alone (dashed lines). To assess the contribution of each party in collaborative mode, we analyze partial sketches with only agent-made strokes (pink) or user-made strokes (green), resulting in a significant reduction in recognizability. This suggests that both user and agent contribute meaningfully to the recognizability of the complete sketch.

5.4. Chat-Based Sketch Editing

We next demonstrate the effectiveness of our method in performing interactive text-based sketch editing within a chat dialogue, where the input to the agent combines both text and images. Inspired by [92], we explore edits that involve spatial reasoning and object relations. We focus on three object categories: outdoor, indoor, and animals, with three objects each, and design editing prompts to add objects to the input sketches. For outdoor and indoor objects, we specify relative locations of added concepts, e.g., "left to", "on top of" (see Fig. 14 left). For the animals category, we tasked



(A) Sketching interface (B) Collaborative user study results

Figure 13. Collaborative sketching evaluation measured using CLIP classification. Sketches created collaboratively (blue) approaching those made solely by users (dashed lines). In collaborative sketches, keeping agent-only strokes (pink) or user-only strokes (green) significantly reduces recognizability.

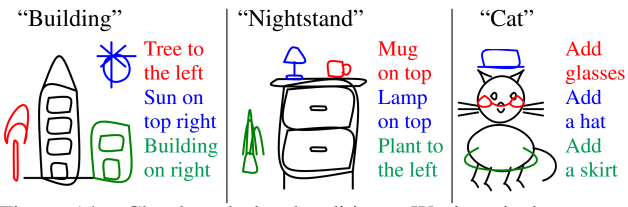


Figure 14. Chat-based sketch editing. We iteratively prompt SketchAgent to add objects to sketches through chat dialogues.

the agent with adding accessories to each animal without guidance on their exact placement, testing its ability to infer placement based on semantics (e.g., placing a hat on a head (see Fig. 14 right). The full list of object and editing instructions is provided in the Appendix. We produced a total of 54 sketches. Evaluating the edited sketches reveals that SketchAgent correctly follows instructions 92% of the time, with 94% accuracy for specified relations and 88% accuracy for inferred semantic relations.

6. Ablation

We evaluate the impact of each component of our method by systematically removing them and measuring sketch recognition rates as detailed in 5.1. We assess the effects of removing the system prompt, omitting the CoT process (i.e., excluding thinking tags and 'think step-by-step' instructions), and modifying ICL (the complete sketch example provided in the user prompt). When modifying ICL, we use a correctly formatted single-stroke example instead of the complete sketch, as fully removing ICL results in outputs that do not follow the expected format making them unparsable. The results in Table 2 show that the full SketchAgent pipeline achieves the highest performance, highlighting the importance of each component. Interestingly, not providing a complete sketch example significantly reduces performance. We additionally ablate the impact of the grid resolution, by varying the resolution from 10 to 100 (see Tab. 2, bottom). Extremely low resolutions degrade performance, while mid-level resolutions outperform 100×100.

	w/o System Prompt	w/o CoT	Modified ICL	SketchAgent (full)
Top1	0.20 ± 0.04	0.14 ± 0.02	0.07 ± 0.02	0.23 ± 0.04
Top5	0.42 ± 0.03	0.29 ± 0.04	0.16 ± 0.03	0.43 ± 0.06

Grid Size	10×10	25×25	$\mathbf{50 \times 50}$	75×75	100×100
Top1 / Top5	0.14 / 0.28	0.19 / 0.42	**0.23 / 0.43**	0.23 / 0.41	0.19 / 0.37

Table 2. Ablation study. Average Top-1 and Top-5 CLIP recognition accuracy. Top: We systematically remove each component in our pipeline, showcasing all components contribute to the agent's full performance. Bottom: Grid resolution ablation.

7. Limitations and Future Work

SketchAgent has several limitations. First, it is constrained by the priors of the backbone model, primarily optimized for text rather than visual content. As a result, the agent often produces rich textual descriptions of object parts but struggles to convert these into effective sketching actions, resulting in overly abstract and unrecognizable outputs. For example, in Fig. 15A, the agent effectively describes key parts of a unicorn (e.g., the horn), but the sketch is unrecognizable. This constraint also impacts the depiction of human figures (Fig. 15B). While distinctive features (e.g., Frida Kahlo's eyebrows or Michael Jordan's dunk) may be captured well in language, the resulting sketches are overly simple, with an amateur style, lacking expressivity. We expect this issue to improve as future models advance in vision capabilities. Lastly, the agent may struggle with drawing letters and numbers. This could be improved in future work by providing relevant in-context examples.

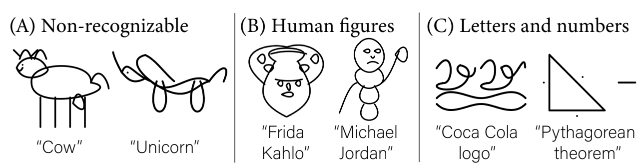


Figure 15. Limitations. Sketches of complex concepts (A) and human figures (B) appear too abstract and unrecognizable with non-professional style. (C) Fail to depict letters and numbers.

8. Conclusions

We presented a method for language-driven, sequential sketch generation, that can produce versatile sketches in real-time and meaningfully engage in collaborative sketching sessions with humans. We show that the prior knowledge embedded in pretrained multimodal LLMs can be effectively leveraged for sketch generation through an intuitive sketching language and a grid canvas, without requiring additional training or fine-tuning. We hope our work represents a meaningful step toward developing general-purpose sketching systems with the potential to enhance human-computer communication and computer-aided ideation.

9. Acknowledgements

We thank Yuval Alaluf, Hila Chefer, Assaf Ben Kish, Joanna Materzynska, Rinon Gal, Elad Richardson, Arnav Verma, and Ellie Arar for providing feedback on early versions of our manuscript. We are also grateful to Justin Yang, who developed the initial Photodraw application and assisted with the interactive human study setup, and to Dingning Cao, who contributed to the SketchAgent interface and application. Special thanks to Yarden Frenkel for his insights, early explorations, and inspiring discussions. This work was partially supported by NSF CAREER #2047191, NSF DRL #2400471, Stanford Human Centered AI Institute Hoffman-Yee Grant, Hyundai Motor Company, ARL grant W911NF-18-2-021, the Zuckerman STEM Leadership Program, and the Viterbi Fellowship. The sponsors had no role in the experimental design or analysis, the decision to publish, or manuscript preparation. The authors have no competing interests to report.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. Red Hook, NY, USA, 2024. Curran Associates Inc. 2, 3

[2] Anthropic. Claude. https://www.anthropic.com/claude, 2023. 2, 3, 5, 6

[3] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. Style and abstraction in portrait sketching. *ACM Trans. Graph.*, 32(4), 2013. 2

[4] James Betker, Gabriel Goh, Li Jing, † TimBrooks, Jianfeng Wang, Linjie Li, † LongOuyang, † JuntangZhuang, † JoyceLee, † YufeiGuo, † WesamManassra, † PrafullaDhariwal, † CaseyChu, † YunxinJiao, and Aditya Ramesh. Improving image generation with better captions. 2023. 3

[5] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: a competitive sketching ai agent. so you think you can sketch? *ACM Trans. Graph.*, 39:166:1–166:15, 2020. 1, 3

[6] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Jorma Laaksonen, and Michael Felsberg. Doodleformer: Creative sketch drawing with transformers. *ECCV*, 2022. 1, 2, 3

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901. Curran Associates, Inc., 2020. 2, 3, 5

[8] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 3

[9] Mu Cai, Zeyi Huang, Yuheng Li, Haohan Wang, and Yong Jae Lee. Delving into LLMs' visual understanding ability using SVG to bridge image and text, 2024. 2, 3

[10] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 2

[11] Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation, 2020. 3

[12] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7915–7925, 2022. 2

[13] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1254–1262, 2024. 3

[14] Changwoon Choi, Jaeah Lee, Jaesik Park, and Young Min Kim. 3doodle: Compact abstraction of objects with 3d strokes. *ACM Trans. Graph.*, 43(4), 2024. 3

[15] Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. Visionllama: A unified llama backbone for vision tasks, 2024. 2, 3

[16] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 632–647. Springer, 2020. 1

[17] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, Sanat Moningi, and Brian Magerko. Drawing apprentice: An enactive co-creative agent for artistic collaboration. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, page 185–186, New York, NY, USA, 2015. Association for Computing Machinery. 3

[18] Vanessa de Andrade, Sofia Freire, Monica Baptista, and Yael Shwartz. Drawing as a space for social-cognitive interaction. *Education Sciences*, 12:45, 2022. 3

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 3

[20] Sivan Doveh, Shaked Perek, M Jehanzeb Mirza, Wei Lin, Amit Alfassy, Assaf Arbelle, Shimon Ullman, and Leonid Karlinsky. Towards multimodal in-context learning for vision & language models. *arXiv preprint arXiv:2403.12736*, 2024. 3

[21] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31(4), 2012. 2

[22] Ziv Epstein, Aaron Hertzmann, Laura Mariah Herman, Robert Mahari, Morgan R. Frank, Matthew Groh, Hope Schroeder, Amy Smith, Memo Akten, Jessica Fjeld, Hany Farid, Neil Leach, Alex Pentland, and Olga Russakovsky. Art and the science of generative ai. *Science*, 380:1110 – 1111, 2023. 1

[23] Judy Fan, Wilma A. Bainbridge, Rebecca Chamberlain, and Jeffrey D. Wammes. Drawing as a versatile cognitive tool. *Nature Reviews Psychology*, 2:556 – 568, 2023. 2

[24] Judith E. Fan, Monica Dinculescu, and David Ha. collabdraw: An environment for collaborative sketching with an artificial agent. In *Proceedings of the 2019 Conference on Creativity and Cognition*, page 556–561, New York, NY, USA, 2019. Association for Computing Machinery. 3, 7

[25] Judith E. Fan, Robert D. Hawkins, Mike Wu, and Noah D. Goodman. Pragmatic Inference and Visual Abstraction Enable Contextual Flexibility During Visual Communication. *Computational Brain & Behavior*, 3(1):86–101, 2020. 3

[26] Judith E Fan, Wilma A Bainbridge, Rebecca Chamberlain, and Jeffrey D Wammes. Drawing as a versatile cognitive tool. *Nature Reviews Psychology*, 2(9):556–568, 2023. 1

[27] Logan Fiorella and Shelbi Kuhlmann. Creating drawings enhances learning by teaching. *Journal of Educational Psychology*, 112(4):811, 2020. 1

[28] Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, and Jon Wetzel. Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 3(4):648–666, 2011. 1

[29] Kevin Frans, L. B. Soros, and Olaf Witkowski. Clipdraw: exploring text-to-drawing synthesis through language-image encoders. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 1, 3

[30] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024. 4

[31] Rinon Gal, Yael Vinker, Yuval Alaluf, Amit Bermano, Daniel Cohen-Or, Ariel Shamir, and Gal Chechik. Breathing life into sketches using text-to-video priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4336, 2024. 3

[32] Yaroslav Ganin, Tejas D. Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. *ArXiv*, abs/1804.01118, 2018. 3

[33] Yaroslav Ganin, Sergey Bartunov, Yujia Li, Ethan Keller, and Stefano Saliceti. Computer-aided design as language. In *Neural Information Processing Systems*, 2021. 3

[34] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5174–5183, 2020. 2

[35] Simon Garrod, Nicolas Fay, John Lee, Jon Oberlander, and Tracy MacLeod. Foundations of representation: Where might graphical symbol systems come from? *Cognitive Science*, 31(6):961–987, 2007. 3

[36] Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. Creative sketch generation. *arXiv preprint arXiv:2011.10039*, 2020. 1

[37] Songwei Ge, Vedanuj Goswami, Larry Zitnick, and Devi Parikh. Creative sketch generation. In *International Conference on Learning Representations*, 2021. 2, 6

[38] Hannie Gijlers, Armin Weinberger, Alieke Mattia van Dijk, Lars Bollen, and Wouter van Joolingen. Collaborative drawing on a shared digital canvas in elementary science education: The effects of script and task awareness support. *International Journal of Computer-Supported Collaborative Learning*, 8(4):427–453, 2013. 3

[39] Gabriela Goldschmidt. Serial sketching: visual problem solving in designing. *Cybernetics and System*, 23(2):191–219, 1992. 1

[40] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia Pont, Frédo Durand, and Adrien Bousseau. Opensketch: a richly-annotated dataset of product design sketches. *ACM Trans. Graph.*, 38(6), 2019. 2

[41] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14375–14385, 2024. 2

[42] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017. 1, 2, 3, 6, 7

[43] Yucheng Han, China. Xiaoyan Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. Chartllama: A multimodal llm for chart understanding and generation. *ArXiv*, abs/2311.16483, 2023. 2, 3

[44] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024. 3

[45] Francisco Javier Ibarrola, Tomas Lawton, and Kazjon Grace. A collaborative, interactive and context-aware drawing agent for co-creative design. *IEEE Transactions on Visualization and Computer Graphics*, 30:5525–5537, 2022. 3

[46] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1911–1920, 2023. 1, 3

[47] Jongejan Jonas, Rowley Henry, Kawashima Takashi, Kim Jongmin, and Fox-Gieg Nick. The Quick, Draw! - A.I. Experiment, 2016. 2, 3, 5, 6, 7

[48] David Kaiser. *Drawing theories apart: The dispersion of Feynman diagrams in postwar physics*. University of Chicago Press, 2019. 1

[49] Pegah Karimi, Jeba Rezwana, Safat Siddiqui, Mary Lou Maher, and Nasrin Dehbozorgi. Creative sketching partner: an analysis of human-ai co-creativity. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, page 221–230, New York, NY, USA, 2020. Association for Computing Machinery. 3

[50] Pegah Karimi, Jeba Rezwana, Safat Siddiqui, Mary Lou Maher, and Nasrin Dehbozorgi. Creative sketching partner: an analysis of human-ai co-creativity. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, page 221–230, New York, NY, USA, 2020. Association for Computing Machinery. 3

[51] Andrew Kyle Lampinen, Stephanie C. Y. Chan, Aaditya K. Singh, and Murray Shanahan. The broader spectrum of in-context learning, 2024. 2

[52] Paul Laseau. *Graphic thinking for architects and designers*. John Wiley & Sons, 2000. 2

[53] Tomas Lawton, Kazjon Grace, and Francisco J Ibarrola. When is a tool a tool? user perceptions of system agency in human–ai co-creative drawing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, page 1978–1996, New York, NY, USA, 2023. Association for Computing Machinery. 3

[54] Tomas Lawton, Francisco J Ibarrola, Dan Ventura, and Kazjon Grace. Drawing with reframer: Emergence and control in co-creative ai. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, page 264–277, New York, NY, USA, 2023. Association for Computing Machinery. 3

[55] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. Shadowdraw: real-time user guidance for freehand drawing. In *ACM SIGGRAPH 2011 Papers*, New York, NY, USA, 2011. Association for Computing Machinery. 3

[56] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2, 3

[57] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1403–1412. IEEE, 2019. 2

[58] Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15, 2020. 1, 3

[59] Yi Li, Yi-Zhe Song, Timothy M. Hospedales, and Shaogang Gong. Free-hand sketch synthesis with deformable stroke models. *CoRR*, abs/1510.02644, 2015. 1, 2

[60] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, and Ming-Hsuan Yang. Im2pencil: Controllable pencil illustration from photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1525–1534, 2019. 2

[61] Hangyu Lin, Yanwei Fu, Yu-Gang Jiang, and X. Xue. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6766, 2020. 3

[62] Difan Liu, Matthew Fisher, Aaron Hertzmann, and Evangelos Kalogerakis. Neural strokes: Stylized line drawing of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14204–14213, 2021. 2

[63] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[64] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916. Curran Associates, Inc., 2023. 2, 3

[65] Bria Long, Judith Fan, Holly Huey, Zixian Chai, and Michael Frank. Parallel developmental changes in children's production and recognition of line drawings of visual concepts. *Nature Communications*, 15, 2024. 6

[66] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[67] William P. McCarthy, Justin Matejka, Karl D.D. Willis, Judith E. Fan, and Yewen Pu. Communicating design intent using drawing and text. In *Proceedings of the 16th Conference on Creativity & Cognition*, page 512–519, New York, NY, USA, 2024. Association for Computing Machinery. 3

[68] John FJ Mellor, Eunbyung Park, Yaroslav Ganin, Igor Babuschkin, Tejas Kulkarni, Dan Rosenbaum, Andy Ballard, Theophane Weber, Oriol Vinyals, and SM Eslami. Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007*, 2019. 3

[69] Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. *Advances in Neural Information Processing Systems*, 34:7153–7166, 2021. 3

[70] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024. 3

[71] Kushin Mukherjee, Holly Huey, Xuanchen Lu, Yael Vinker, Rio Aguina-Kang, Ariel Shamir, and Judith Fan. Seva: Leveraging sketches to evaluate alignment between human and machine visual abstraction. In *Advances in Neural Information Processing Systems*, 2023. 2

[72] Omar W Nasim. *Observing by hand: sketching the nebulae in the nineteenth century*. University of Chicago Press, 2019. 1

[73] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 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*, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery. 3

[74] OpenAI. Gpt-4 technical report, 2024. 2, 3, 4, 6

[75] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. 2, 3

[76] Yonggang Qi, Guoyao Su, Pinaki Nath Chowdhury, Mingkang Li, and Yi-Zhe Song. Sketchlattice: Latticed representation for sketch manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 953–961, 2021. 2

[77] Shuwen Qiu, Sirui Xie, Lifeng Fan, Tao Gao, Song-Chun Zhu, and Yixin Zhu. Emergent graphical conventions in a visual communication game. *arXiv preprint arXiv:2111.14210*, 2021. 3

[78] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 1, 3, 6

[79] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), 2020. 3

[80] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 3

[81] Leo Sampaio Ferraz Ribeiro, Tu Bui, John P. Collomosse, and Moacir Antonelli Ponti. Sketchformer: Transformer-based representation for sketched structure. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14141–14150, 2020. 3

[82] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2, 3, 6

[83] Daniel Rose, Vaishnavi Himakunthala, Andy Ouyang, Ryan He, Alex Mei, Yujie Lu, Michael Saxon, Chinmay Sonar, Diba Mirza, and William Yang Wang. Visual chain of thought: bridging logical gaps with multimodal infillings. *arXiv preprint arXiv:2305.02317*, 2023. 3

[84] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 3

[85] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4), 2016. 2

[86] Peter Schaldenbrand, James McCann, and Jean Oh. Frida: A collaborative robot painter with a differentiable, real2sim2real planning environment. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11712–11718, 2023. 3

[87] Peter Schaldenbrand, Gaurav Parmar, Jun-Yan Zhu, James McCann, and Jean Oh. Cofrida: Self-supervised fine-tuning for human-robot co-painting. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024. 3

[88] Donald A Schon and Vincent DeSanctis. The reflective practitioner: How professionals think in action, 1986. 2

[89] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 3

[90] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024. 3

[91] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *arXiv preprint arXiv:2403.16999*, 2024. 3

[92] Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14410–14419, 2024. 3, 7

[93] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Learning to sketch with shortcut cycle consistency, 2018. 2

[94] Guoyao Su, Yonggang Qi, Kaiyue Pang, Jie Yang, Yi-Zhe Song, and CVSSP SketchX. Sketchhealer: A graph-to-sequence network for recreating partial human sketches. In *BMVC*, page 5, 2020. 2

[95] Yanpeng Sun, Qiang Chen, Jian Wang, Jingdong Wang, and Zechao Li. Exploring effective factors for improving visual in-context learning. *arXiv preprint arXiv:2304.04748*, 2023. 3

[96] Ivan E. Sutherland. *Sketchpad—a man-machine graphical communication system*, page 391–408. Association for Computing Machinery, New York, NY, USA, 1998. 3

[97] Gemini Team. Gemini: A family of highly capable multimodal models, 2024. 2, 3

[98] Jakob Tholander and Martin Jonsson. Design ideation with ai - sketching, thinking and talking with generative machine learning models. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, page 1930–1940, New York, NY, USA, 2023. Association for Computing Machinery. 1

[99] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024. 4

[100] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. 3

[101] Barbara Tversky. What do sketches say about thinking? *AAAI Spring Symp. Sketch Understanding Worksh.*, 2002. 2

[102] Barbara Tversky. Visualizing thought. In *Handbook of human centric visualization*, pages 3–40. Springer, 2013. 1

[103] Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. Sketches for design and design of sketches. *Human Behaviour in Design: Individuals, Teams, Tools*, pages 79–86, 2003. 1

[104] Russell Tytler, Vaughan Prain, George Aranda, Joseph Ferguson, and Radhika Gorur. Drawing to reason and learn in science. *Journal of Research in Science Teaching*, 57(2): 209–231, 2020. 3

[105] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.*, 41(4), 2022. 1, 3, 6

[106] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4146–4156, 2023. 3, 6

[107] Jiawei Wang and Changjian Li. Contextseg: Sketch semantic segmentation by querying the context with attention. *arXiv preprint arXiv:2311.16682*, 2023. 6

[108] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 2, 5

[109] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C. Olsen. Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Comput. Graph.*, 36:740–753, 2012. 2

[110] World Wide Web Consortium (W3C). *Scalable Vector Graphics (SVG)*, 1999. 3

[111] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 3

[112] Rong Wu, Wanchao Su, Kede Ma, and Jing Liao. Iconshop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 3

[113] Chufeng Xiao, Wanchao Su, Jing Liao, Zhouhui Lian, Yi-Zhe Song, and Hongbo Fu. Differsketching: How differently do people sketch 3d objects? *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2022)*, 41 (4):1–16, 2022. 2

[114] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[115] XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. In *Advances in Neural Information Processing Systems*, pages 15869–15889. Curran Associates, Inc., 2023. 3, 6

[116] Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. Svgdreamer: Text guided svg generation with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4546–4555, 2024. 3, 6, 7

[117] Peng Xu, Timothy M. Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey and a toolbox, 2020. 2

[118] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *ArXiv*, abs/2309.17421, 2023. 2

[119] Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. *arXiv preprint arXiv:2310.08541*, 2023. 3

[120] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2019. 2

[121] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Unpaired portrait drawing generation via asymmetric cycle mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8217–8225, 2020. 2

[122] C. Zhang, Weijie Wang, Paul Pangaro, Nikolas Martelaro, and Daragh Byrne. Generative image ai using design sketches as input: Opportunities and challenges. *Proceed-*

ings of the 15th Conference on Creativity and Cognition, 2023. 1

[123] Jiahao Zhang, Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Instruct me more! random prompting for visual in-context learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2597–2606, 2024. 3

[124] Peiying Zhang, Nanxuan Zhao, and Jing Liao. Text-to-vector generation with neural path representation. *ACM Trans. Graph.*, 43(4), 2024. 3

[125] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36: 17773–17794, 2023. 3

[126] Zhengming Zhang, Xiaoming Deng, Jinyao Li, Yukun Lai, Cuixia Ma, Yongjin Liu, and Hongan Wang. Stroke-based semantic segmentation for scene-level free-hand sketches. *Vis. Comput.*, 39(12):6309–6321, 2022. 6

[127] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 3

[128] Yixiao Zheng, Kaiyue Pang, Ayan Das, Dongliang Chang, Yi-Zhe Song, and Zhanyu Ma. Creativeseg: Semantic segmentation of creative sketches. *IEEE Transactions on Image Processing*, 33:2266–2278, 2024. 6

[129] Tao Zhou, Chen Fang, Zhaowen Wang, Jimei Yang, Byungmoon Kim, Zhili Chen, Jonathan Brandt, and Demetri Terzopoulos. Learning to sketch with deep q networks and demonstrated strokes. *ArXiv*, abs/1810.05977, 2018. 2, 3