MagicQuill: An Intelligent Interactive Image Editing System

Zichen Liu[♡,1,2], Yue Yu[♡,1,2], Hao Ouyang[2], Qiuyu Wang[2],
Ka Leong Cheng[1,2], Wen Wang[3,2], Zhiheng Liu[4], Qifeng Chen[†,1], Yujun Shen[†,2]

[1]HKUST, [2]Ant Group, [3]ZJU, [4]HKU

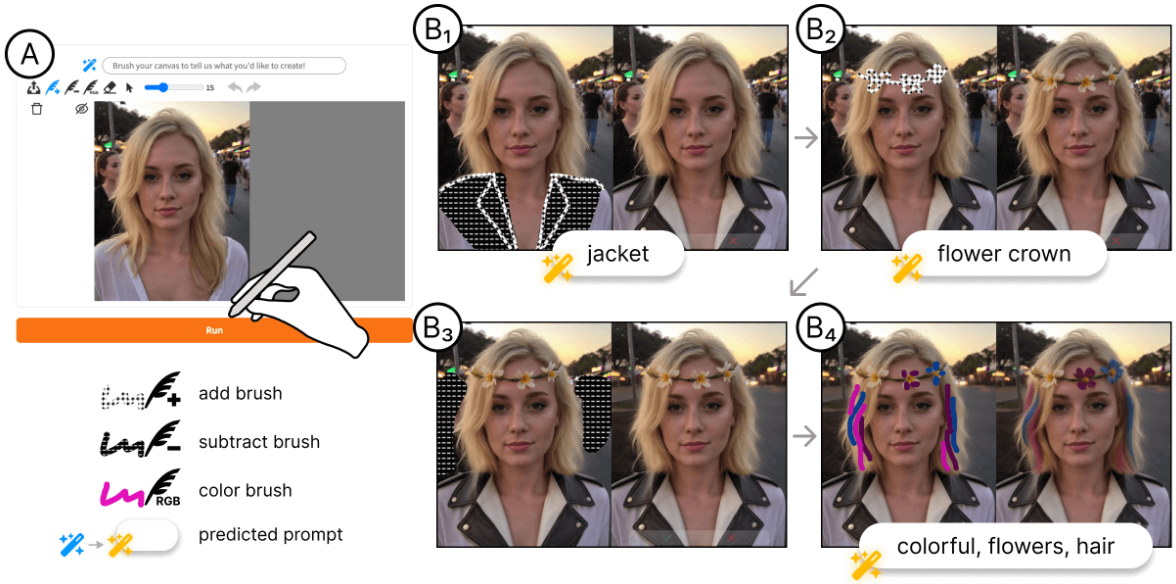


Figure 1. MagicQuill is an intelligent and interactive image editing system built upon diffusion models. Users seamlessly edit images using three intuitive brushstrokes: add, subtract, and color (A). A MLLM dynamically predicts user intentions from their brush strokes and suggests contextual prompts (B1-B4). The examples demonstrate diverse editing operations: to generate a jacket from clothing contour (B1), add a flower crown from head sketches (B2), remove background (B3), and apply color changes to the hair and flowers (B4).

Abstract

As a highly practical application, image editing encounters a variety of user demands and thus prioritizes excellent ease of use. In this paper, we unveil MagicQuill, an integrated image editing system designed to support users in swiftly actualizing their creativity. Our system starts with a streamlined yet functionally robust interface, enabling users to articulate their ideas (e.g., inserting elements, erasing objects, altering color, etc.) with just a few strokes. These interactions are then monitored by a multimodal large language model (MLLM) to anticipate user intentions in real time, bypassing the need for prompt entry. Finally, we apply the powerful diffusion prior, enhanced by a carefully learned two-branch plug-in module, to process the editing request with precise control. Please visit the project page to try out our system.

1. Introduction

Performing precise and efficient edits on digital photographs remains a significant challenge, especially when aiming for nuanced modifications. As shown in Fig. 1, consider the task of editing a portrait of a lady where specific alterations are desired: converting a shirt to a custom-designed jacket, adding a flower crown at an exact position with a well-designed shape, dyeing portions of her hair in particular colors, and removing certain parts of the background to refine her appearance. Despite the rapid advancements in diffusion models [7, 12, 17, 22, 40, 42–44, 52, 71, 77] and recent attempts to enhance control [23, 26, 53, 78], achieving such fine-grained and precise edits continues to pose difficulties, typically due to a lack of intuitive interfaces and models for fine-grained control.

The challenges highlight the critical need for interactive editing systems that facilitate precise and efficient modifications. An ideal solution would empower users to specify

♡Equal contribution. †Corresponding author.

what they want to edit, ***where*** to apply the changes, and ***how*** the modifications should appear, all within a ***user-friendly*** interface that streamlines the editing process.

We aim to develop the first robust, ***open-source***, interactive precise image editing system to make image editing easy and efficient. Our system seamlessly integrates three core modules: the **Editing Processor**, the **Painting Assistor**, and the **Idea Collector**. The Editing Processor ensures a high-quality, controllable generation of edits, accurately reflecting users' editing intentions in color and edge adjustments. The Painting Assistor enhances the ability of the system to predict and interpret the users' editing intent. The Idea Collector serves as an intuitive interface, allowing users to input their ideas quickly and effortlessly, significantly boosting the editing efficiency.

The Editing Processor implements two kinds of brushstroke-based guidance mechanisms: scribble guidance for structural modifications (e.g., adding, detailing, or removing elements) and color guidance for modification of color attributes. Inspired by ControlNet [75] and BrushNet [26], our control architecture ensures precise adherence to user guidance while preserving unmodified regions. Our Painting Assistor reduces the repetitive process of typing text prompts, which disrupts the editing workflow and creates a cumbersome transition between prompt input and image manipulation. It employs an MLLM to interpret brushstrokes and automatically predicts prompts based on image context. We call this novel task Draw&Guess. We construct a dataset simulating real editing scenarios for fine-tuning to ensure the effectiveness of the MLLM in understanding user intentions. This enables a continuous editing workflow, allowing users to iteratively edit images without manual prompt input. The Idea Collector provides an intuitive interface compatible with various platforms including Gradio and ComfyUI, allowing users to draw with different brushes, manipulate strokes, and perform continuous editing with ease.

We present a comprehensive evaluation of our interactive editing framework. Through qualitative and quantitative analyses, we demonstrate that our system significantly improves both the precision and efficiency of performing detailed image edits compared to existing methods. Our Editing Processor achieves superior edge alignment and color fidelity compared to baselines like SmartEdit [23] and BrushNet [26]. The Painting Assistor exhibits superior user intent interpretation capabilities compared to state-of-the-art MLLMs, including LLaVA-1.5 [35], LLaVA-Next [34], and GPT-4o [24]. User studies indicate that the Idea Collector significantly outperforms baseline interfaces in all aspects of system usability.

By leveraging advanced generative models and a user-centric design, our interactive editing framework significantly reduces the time and expertise required to perform detailed image edits. By addressing the limitations of current image editing tools and providing an innovative solution that enhances both precision and efficiency, our work advances the field of digital image manipulation. Our framework opens possibilities for users to engage creatively with image editing, achieving their goals easily and effectively.

2. Related Works

2.1. Image Editing

Image editing involves modifying the visual appearance, structure, or elements of an existing image [22]. Recent breakthroughs in diffusion models [20, 50, 54] have significantly advanced visual generation tasks, outperforming GAN-based models [18] in terms of image editing capabilities. To enable control and guidance in image editing, a variety of approaches have emerged, leveraging different modalities such as textual instructions [3, 4, 7, 9, 13, 16, 17, 27, 28, 52, 66, 67, 74], masks [23, 26, 53, 62, 78], layouts [12, 37, 77], segmentation maps [40, 71], strokes [41, 70], references [8, 36, 38, 55], and point-dragging interfaces [42–44]. Despite these advances, these methods often fall short when precise modifications at the regional level are required, such as alterations to object shape, color, and other details. Among the various methods, sketch-based editing approaches [25, 29, 39, 48, 65, 69, 73] offer users a more intuitive and precise means of interaction. However, the current methods remain limited by the accuracy of the text signals input alongside the sketches, making it challenging to precisely control the information of the editing areas, such as color. To achieve precise control, we introduce two types of local guidance based on brushstrokes: scribble and color, thereby enabling fine-grained control over shape and color at the regional level.

2.2. MLLMs for Image Editing

Multi-modal large language models (MLLMs) extend LLMs to process both text and image content [19], enabling text-to-image generation [11, 32, 58, 59], prompt-refinement [68, 72], and image quality evaluation [57].

In the area of image editing, MLLMs have demonstrated significant potential. MGIE [15] enhances instruction-based image editing by using MLLMs to generate more expressive, detailed instructions. SmartEdit [23] leverages MLLM for better understanding and reasoning towards complex instruction. FlexEdit [61] integrates MLLM to understand image content, masks, and textual instructions. GenArtist [64] uses an MLLM agent to decompose complex tasks, guide tool selection, and enable systematic image generation, editing, and self-correction with step-by-step verification. Our system extends this line of research by introducing a more intuitive approach, utilizing MLLM

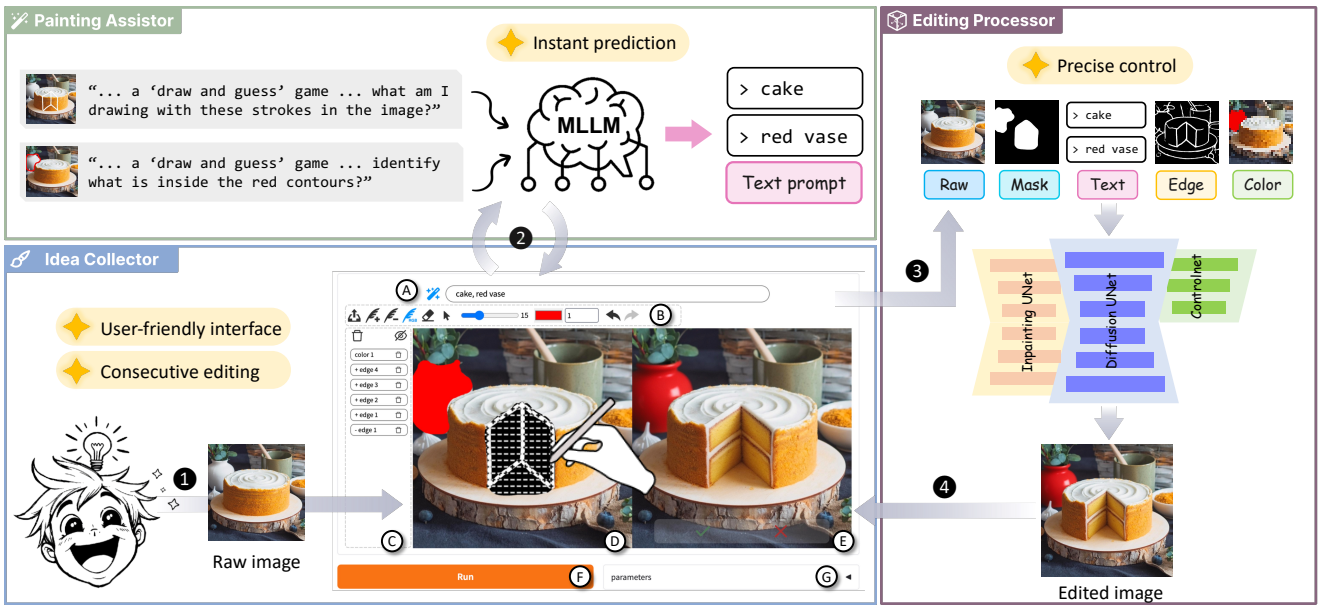


Figure 2. System framework consisting of three integrated components: an **Editing Processor** with dual-branch architecture for controllable image inpainting, a **Painting Assistor** for real-time intent prediction, and an **Idea Collector** offering versatile brush tools. This design enables intuitive and precise image editing through brushstroke-based interactions.

to simplify the editing process. Specifically, it directly integrates the image context with user-input strokes to infer and translate the editing intentions, thereby automatically generating the necessary prompts without requiring repeated user input. This innovative task, which we term Draw&Guess, facilitates a continuous editing workflow, enabling users to iteratively refine images with minimal manual intervention.

2.3. Interactive Support for Image Generation

Interactive support enhances the performance and usability of generative models through human-in-the-loop collaboration [31]. Recent works have focused on making prompt engineering more user-friendly through techniques like image clustering [5, 14] and attention visualization [63].

Despite advancements in interactive support, a key challenge remains in bridging the gap between verbal prompts and visual output. While systems like PromptCharm [63] and DesignPrompt [45] use inpainting for interactive image editing, these tools typically offer only coarse-grained control over element addition and removal, requiring users to brush over areas before generating objects within those regions. Furthermore, users must manually input prompts to specify the objects they wish to generate. Our approach addresses these limitations by introducing fine-grained image editing through the use of brushstrokes. Additionally, we incorporate a multimodal large language model (MLLM) that provides on-the-fly assistance by interpreting user intentions and suggesting prompts in real-time, thereby reducing cognitive load and enhancing overall usability.

3. System Design

Our system is structured around three key aspects: **Editing Processor** with strong generative prior, **Painting Assistor** with instant intent prediction, and **Idea Collector** with a user-friendly interface. An overview of our system design is presented in Fig. 2.

Our system introduces brushstroke-based control signals to give intuitive and precise control. These signals allow users to express their editing intentions by simply drawing what they envision. We designed two types of brushes, scribble and color, to accurately manipulate the edited image. The scribble brushes, **add brush** and **subtract brush**, aim to provide precise structural control by operating on the edge map of the original image. The **color brush** works with downsampled color blocks to enable fine-grained color manipulation of specific regions. Fig. 3 illustrates the workflow to convert the user hand-drawn input signal into control condition for faithfully inpainting the target editing area. Inspired by Ju et al. [26], Zhang et al. [75], we employ two additional branches to the latent diffusion framework [50], with the inpainting branch giving content-aware per-pixel guidance for the re-generation of the editing area, and the control branch providing structural guidance. The model architecture is illustrated in Fig. 4. Further details will be discussed in Sec. 3.1.

To reduce the cognitive load for users to input appropriate prompts at every stage of editing, our system integrates a MLLM [33] as the Painting Assistor. This component analyzes user brushstrokes to deduce the editing intention

based on the image context, thereby automatically suggesting contextually relevant prompts for editing. We have named this innovative task Draw&Guess. To effectively prepare the MLLM for Draw&Guess, we designed a dataset construction method to simulate user hand-drawn editing scenarios and acquire ground truth for Draw&Guess. We fine-tuned a dedicated LLaVA [35] model, achieving instant prompt guessing with satisfactory accuracy. More specifics will be covered in Sec. 3.2.

Additionally, to provide users with a streamlined, intuitive interface that empowers them to express their ideas for complex image editing tasks with ease, we designed an Idea Collector with a user-friendly interface. The key features of the interface will be outlined in Sec. 3.3.

3.1. Editing Processor

Control Condition from Brushstroke Signal: Let $\mathbf{M}_{add}$ and $\mathbf{M}_{sub}$ denote the binary masks corresponding to add and subtract brush respectively. These masks share the same dimensions as the original image $\mathbf{I}$, where values are set to 1 in regions corresponding to user brush strokes and 0 elsewhere. The subtract brush masks out the edges from the edge map $\mathbf{E}$, which is initially extracted from the original image using a pre-trained CNN f_{CNN}. Conversely, the add brush introduces new edges by setting designated regions to white in the edge map. The resulting modified edge map $\mathbf{E}_{cond}$ serves as the control condition for manipulating geometric structure in the editing processor. This can be formally expressed as

$$\begin{aligned} \mathbf{E} &= f_{CNN}(\mathbf{I}), \\ \mathbf{E}_{sub} &= \mathbf{E} \odot (1 - \mathbf{M}_{sub}), \\ \mathbf{E}_{cond} &= \mathbf{E}_{sub} + \mathbf{M}_{add} \odot (1 - \mathbf{E}_{sub}). \end{aligned} \tag{1}$$

For precise region-specific colorization, we represent each color brush stroke as a tuple $(\mathbf{M}_{color}, \mathbf{c}, \alpha)$, where $\mathbf{M}_{color}$ denotes a binary mask indicating the user-defined stroke region, $\mathbf{c}$ specifies the stroke color, and $\alpha \in [0, 1]$ represents the stroke opacity. The colorization operation can be formally expressed as

$$\mathbf{I}_c = (1 - \alpha \cdot \mathbf{M}_{color}) \odot \mathbf{I} + \alpha \cdot \mathbf{M}_{color} \cdot \mathbf{c}, \tag{2}$$

where the color $\mathbf{c}$ with an alpha blending factor α is applied over a specific region of the image $\mathbf{I}$ defined by the binary mask $\mathbf{M}_{color}$.

To generate the color condition $\mathbf{C}_{cond}$, we first downscale the image $\mathbf{I}_c$ by a factor of 16 using cubic interpolation, followed by upscaling to the original resolution using nearest-neighbor interpolation. This process generated a color block preserving the global color structure while simplifying local details.

The edge condition $\mathbf{E}_{cond}$ and color condition $\mathbf{C}_{cond}$ jointly guide the inpainting process for precise editing control. The editing region, represented by mask $\mathbf{M}$, is

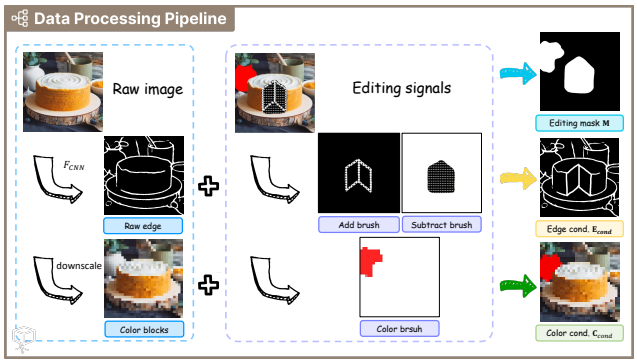


Figure 3. Data processing pipeline. The input image undergoes edge extraction via CNN and color simplification through downscaling. Three editing conditions are then generated based on brush signals: editing mask, edge condition, and color condition, which together provide control for image editing.

obtained by dilating the union of brush regions by p pixels. The masked image $\mathbf{I}_{masked}$ can then be formulated as

$$\begin{aligned} \mathbf{M} &= Grow_p(\mathbf{M}_{add} \cup \mathbf{M}_{sub} \cup \mathbf{M}_{color}), \\ \mathbf{I}_{masked} &= \mathbf{I} \odot (1 - \mathbf{M}). \end{aligned} \tag{3}$$

This expansion accounts for the fact that editing can affect areas surrounding the mask, such as shadows or other adjacent details. By growing the mask, we ensure that these peripheral regions are properly generated, resulting in a more seamless and realistic edit.

Controllable Image Inpainting: The inpainting branch adopts the UNet [26, 51] architecture, incorporating the masked image feature into the pre-trained diffusion network. This branch inputs the concatenated noisy latent at t-th step z_t, masked image latent z_{masked} extracted using VAE [30] from $\mathbf{I}_{masked}$, and downsampled mask $\mathbf{m}$ by cubic interpolation from $\mathbf{M}$. The inpainting branch processes these features, utilizing a trainable clone of the diffusion model, stripped of cross-attention layers to focus solely on the image feature. The extracted features carrying pixel-level information are inserted into each layer of the frozen diffusion model through zero-convolution layers $\mathcal{Z}$ [75]. Given text condition τ, timestep t, let $F(z_t, \tau, t; \Theta)_i$ represents the feature of the i-th layer in the total n layers of the diffusion UNet with parameter Θ. Similarly, let $F^I([z_t, z_{masked}, \mathbf{m}], t; \Theta^I)_i$ represents the output of the i-th layer in the inpainting UNet, where $[\cdot]$ denotes the concatenation operation. This feature insertion can be represented by

$$F(z_t, \tau, t; \Theta)_i \mathrel{+}= w_I \cdot \mathcal{Z}(F^I([z_t, z_{masked}, \mathbf{m}], t; \Theta^I)_i), \tag{4}$$

where w_I is an adjustable hyperparameter that determines the inpainting strength. Equipped with the inpainting branch, the diffusion UNet can fill the masked area in a content-aware manner based on the text prompt.

The control branch aims to introduce conditional generation ability to the diffusion UNet based on condition

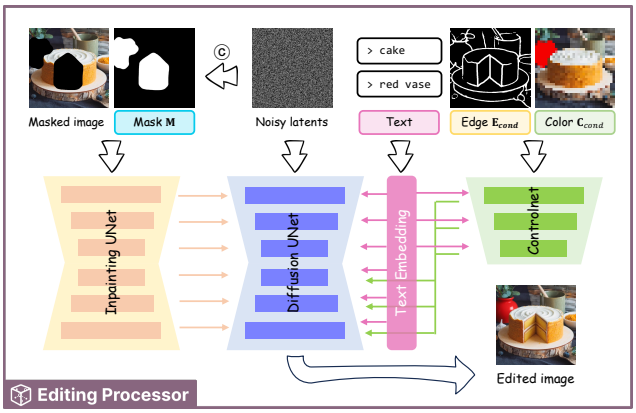


Figure 4. Overview of our Editing Processor. The proposed architecture extends the latent diffusion UNet with two specialized branches: an inpainting branch for content-aware per-pixel inpainting guidance and a control branch for structural guidance, enabling precise brush-based image editing.

$\mathcal{C} = \{\mathbf{E}_{cond}, \mathbf{C}_{cond}\}$. We adopt ControlNet [75] to insert conditional control into the middle and decoder blocks of the diffusion UNet. Let $F^C(z_t, \mathcal{C}, t; \Theta^C)_i$ represent the output of the i-th layer in the ControlNet, the control feature insertion can be formulated as

$$F(z_t, \tau, t; \Theta)_{\lfloor \frac{n}{2} \rfloor + i} += w_C \cdot \mathcal{Z}(F^C(z_t, \mathcal{C}, t; \Theta^C)_i), \quad (5)$$

where w_C is an adjustable hyperparameter that determines the control strength. Both the inpainting and control branches don't alter the weights of the pre-trained diffusion models, enabling it to be a plug-and-play component applicable to any community fine-tuned diffusion models. The control branch is trained using the denoising score matching objective, which can be written as

$$\mathcal{L} = \mathbb{E}_{z_t, t, \epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\left\| \epsilon - \epsilon^c \left(z_t, \mathcal{C}, t; \{\Theta, \Theta^C\} \right) \right\|^2 \right], \quad (6)$$

where ϵ^c is the combination of the denoising U-Net and the ControlNet model.

3.2. Painting Assistor

Prompt formatting: In our system, we implement two types of question answering (Q&A) [2] tasks to facilitate the Draw&Guess. For the add brush, we utilize a prompt structured as follows: *"This is a 'draw and guess' game. I will upload an image containing some strokes. To help you locate the strokes, I will give you the normalized bounding box coordinates of the stokes where their original coordinates are divided by the padded image width and height. The top-left corner of the bounding box is at (x_1, y_1), and the bottom-right corner is at (x_2, y_2). Now tell me in a single word a phrase, what am I trying to draw with these strokes in the image?"* The Q&A output directly serves as the predicted prompt. For the subtract brush, we bypass the Q&A process, as the results demonstrate that prompt-free generation achieves satisfactory results.

For the color brush, the Q&A setup is similar: *"The user will upload an image containing some contours in red color. To help you locate the contour, ... You need to identify what is inside the contours using a single word or phrase."*, (the repetitive part is omitted). The system extracts contour information from the color brush stroke boundaries. The final predicted prompt is generated by combining the stroke's color information with Q&A outputs. To optimize response time, we constrain Q&A responses to concise, single-word or short-phrase formats.

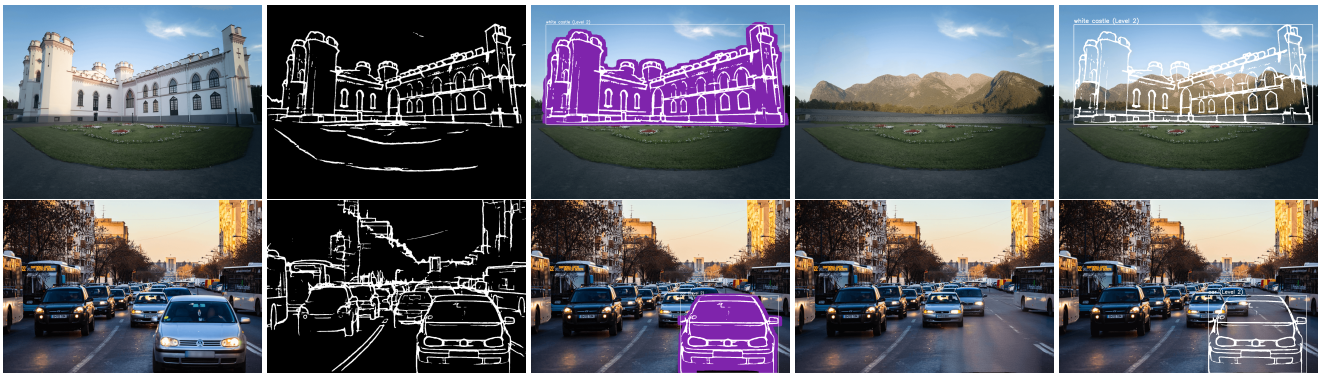
For the color brush Q&A task, accurate object recognition within contours is essential. LLaVA [35] inherently excels in object recognition tasks, making it adept at identifying the content within color brush stroke boundaries. However, the interpretation of add brush strokes poses a significant challenge due to the inherent abstraction of human hand-drawn strokes or sketches. To address this, we find it necessary to construct a specialized dataset to fine-tune LLaVA to better understand and interpret human hand-drawn brush strokes.

Dataset Construction: We selected the Densely Captioned Images (DCI) dataset [60] as our primary source. Each image within the DCI dataset has detailed, multi-granular masks, accompanied by open-vocabulary labels and rich descriptions. This rich annotation structure enables the capture of diverse visual features and semantic contexts.

Step 1: Answer Generation for Q&A. The initial stage involves generating edge maps using PiDiNet [56] from images in the DCI dataset, as shown in Fig. 5b. We calculate the edge density within the masked regions and select the top 5 masks with the highest edge densities, as illustrated in Fig. 5c. The labels corresponding to these selected masks serve as the ground truths for the Q&A. To ensure the model focuses on guessing user intent rather than parsing irrelevant details, we clean the label to keep only noun components, streamlining to emphasize essential elements.

Step 2: Simulating Brushstroke with Edge Overlay. In the second part of the dataset construction, we focus on the five masks identified in the first step. Each mask undergoes random shape expansion to introduce variability. We use the BrushNet [26] model based on the SDXL [47] to perform inpainting on these augmented masks with empty prompt, as shown in Fig. 5d. Subsequently, the edge maps generated earlier are overlaid onto the inpainted areas as in Fig. 5e. These overlay images simulate practical examples of how user hand-drawn strokes might alter an image.

MLLM Fine-Tuning: Our dataset construction method effectively prepares the model to understand and predict user edits, which contains a total of $24,315$ images, categorized under $4,412$ different labels, ensuring a broad spectrum of data for training. To optimize the performance of the MLLM over Draw&Guess, we fine-tuned the LLaVA



(a) Original Image (b) Edge Map (c) Chosen Mask (d) Inpainting Result (e) Edge Overlay

Figure 5. Illustration of dataset construction process. (a) Original images from the DCI dataset; (b) Edge maps extracted from original images; (c) Selected masks (highlighted in purple) with highest edge density; (d) Results after BrushNet inpainting on augmented masked regions; (e) Final results with edge map overlay on selected areas. By overlaying edge maps on inpainted results, we simulate scenarios where users edit images with brush strokes, as the edge maps resemble hand-drawn sketches. The bounding box coordinates of the mask and labels are inherited from the DCI dataset.

model, leveraging the Low-Rank Adaptation (LoRA) [21] technique, allowing the efficient fine-tuning without extensively large dataset. Consistent with the original LLaVA training objectives, our approach aims to maximize the likelihood of the correct labels given the input corpora u, which is defined as

$$\max_{\Theta^{lora}} \sum_{i=1}^{|u|} \log P\left(u_i \mid u_1, \ldots, u_{i-1}; \{\Theta^{pt}, \Theta^{lora}\}\right), \quad (7)$$

where Θ^{pt} and Θ^{lora} are parameters in the pre-trained MLLM and the LoRA respectively.

3.3. Idea Collector

The user interface of `MagicQuill` is designed for an intuitive and streamlined image editing experience, as depicted in Figure 2. The interface is divided into several interactive sections, emphasizing ease of use while providing flexible control over the editing process. The interface comprises several key areas: a *Prompt Area (A)* displaying MLLM-suggested prompts, a *Toolbar (B)* with essential editing tools, *Layer Management (C)* for organizing brush strokes, the main *Canvas (D)* for editing, a *Generated Images area (E)* for previewing results, *Execute Button (F)*, and *Parameter Adjustment (G)*.

4. Experiment

In evaluating our system, we focused on three primary modules: the **Editing Processor**, the **Painting Assistor**, and the **Idea Collector**. First, we assessed the quality of controllable generation provided by the Editing Processor, with particular attention to edge alignment and color fidelity. This evaluation involved analyzing how effectively users could manipulate and achieve desired visual outputs, which ensures the system responds accurately to user's control signal, detailed in Sec. 4.1. Second, We evaluated the Painting Assistor's semantic prediction accuracy using simulated hand-drawn inputs. This assessment was critical for validating the capability of the MLLM in interpreting user intentions, ensuring contextually appropriate suggestions that align with the image semantics. Additionally, we conducted user studies to gather feedback on the system's efficiency improvements and prediction accuracy in real-world scenario, presented in Sec. 4.2. Third, we assessed the usability of the user interfaces across all modules. We decomposes the assessment into four distinct dimensions spanning from operational efficiency to user satisfaction. This multi-dimensional assessment framework enabled systematic comparison with baseline systems while ensuring thorough evaluation of the interface, as shown in Sec. 4.3.

4.1. Controllable Generation

To thoroughly evaluate the controllable generation capabilities of our editing processor, we compared it with four representative baselines from different categories: (1) SmartEdit [23], an instruction-based editing method. We utilize LLaVA-Next [34] to generate the editing instruction; (2) SketchEdit [73], a GAN-based sketch-conditioned method; (3) BrushNet [26], the mask and prompt-guided inpainting method; and (4) a composite baseline combining BrushNet [26] and ControlNet [75]. As illustrated in Fig. 6, the instruction-based method, SmartEdit, tends to produce outputs that are too random, lacking the precision required for accurate editing purposes. Similarly, while BrushNet enables region-specific modifications, it struggles with maintaining predictable detail generation even with ControlNet enhancement, making precise manipulation challenging. In contrast, our model achieves more accurate edge alignment and color fidelity, which we attribute to our specialized design of the inpainting and control branch that emphasizes these aspects.

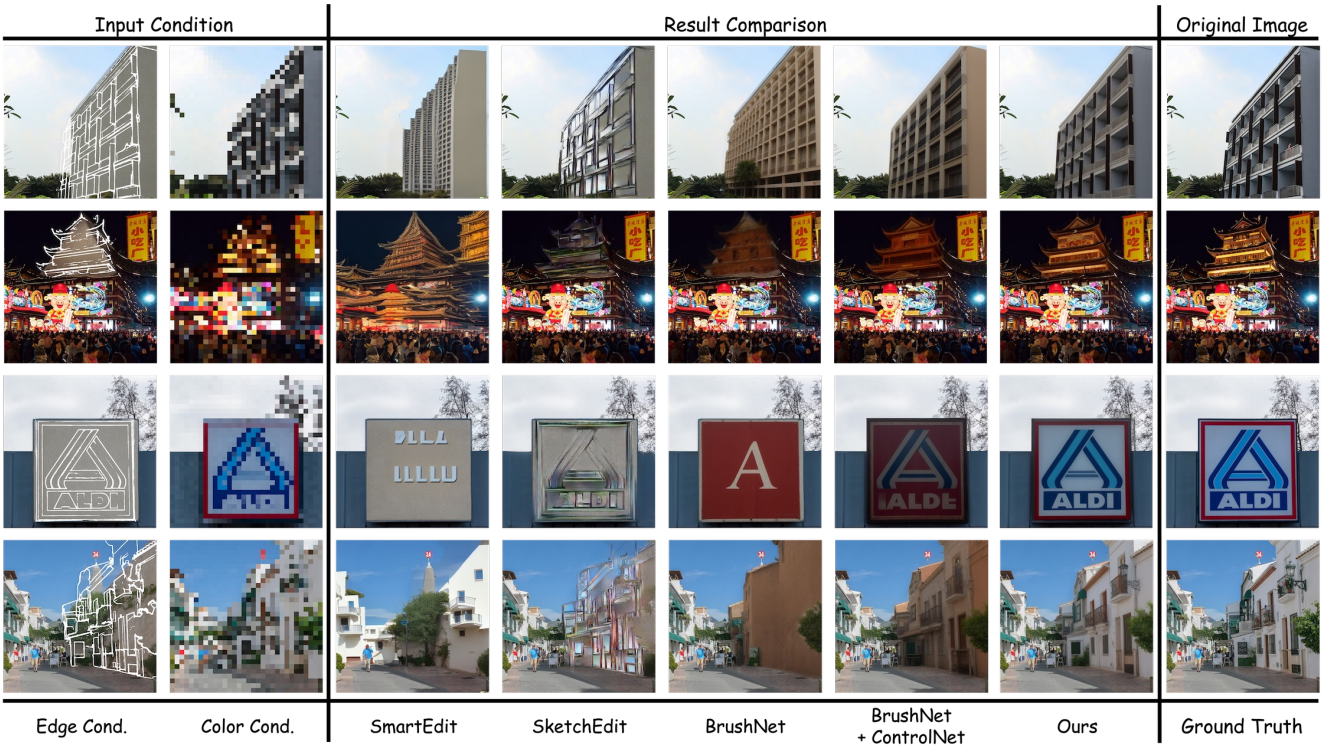


Figure 6. Visual result comparison. The first two columns present the edge and color conditions for editing, while the last column shows the ground truth image that the models aim to recreate. SmartEdit [23] utilizes natural language for guidance, but lacks precision in controlling shape and color, often affecting non-target regions. SketchEdit [73], a GAN-based approach [18], struggles with open-domain image generation, falling short compared to models with diffusion-based generative priors. Although BrushNet [26] delivers seamless image inpainting, it struggles to align edges and colors simultaneously, even with ControlNet [75] enhancement. In contrast, our Editing Processor strictly adheres to both edge and color conditions, achieving high-fidelity conditional image editing.

Table 1. Quantitative results and input condition comparisons between the baselines and ours. Our Editing Processor performs better than the baselines across all metrics, indicating its superiority in controllable generation over edge and color.

Method	Input Condition			LPIPS[76]	PSNR	SSIM
	Text	Edge	Color			
SmartEdit	✓	✗	✗	0.339	16.695	0.561
SketchEdit	✗	✓	✗	0.138	23.288	0.835
BrushNet	✓	✗	✗	0.0817	25.455	0.893
Brush.+Cont.	✓	✓	✓	0.0748	25.770	0.894
Ours	✓	✓	✓	**0.0667**	**27.282**	**0.902**

We conducted a quantitative analysis of our constructed test dataset in Sec. 3.2, which contains 490 images. Our model outperformed the baselines across all key metrics as in Tab. 1. These results demonstrate significant improvements in controllable generation.

We additionally compared two stroke-based editing methods SDEdit [41] and UniPaint [70], and the qualitative results are shown below in Fig. 7.

Figure 7. Visual comparison with stroke-based editing baselines.

4.2. Prediction Accuracy & Efficiency Facilitation

To evaluate the prediction accuracy of the Painting Assistor, we compared it with three state-of-the-art MLLMs: LLaVA-1.5 [35], LLaVA-Next [34], and GPT-4o [24] on our test dataset of 490 images from Sec. 3.2. Each model was prompted with images containing sketches and bounding box coordinates to generate semantic interpretations. The semantic outputs were assessed using three metrics: BERT [10], CLIP [49], and GPT-4 [1] similarity scores, which measure the closeness of the generated descriptions to the ground truth. For GPT-4 similarity, we ask GPT-4 to rate the semantic and visual similarity between the predicted response and the ground truth on a 5-point scale, where 1 means "completely different", 3 means "somewhat related", and 5 means "exactly same".

The evaluation results are presented in Tab. 2, illustrating that our model achieves the highest prediction accuracy among all tested MLLMs. This superior performance indicates that our Painting Assistor more accurately captures and predicts the semantic meanings of user drawings.

To qualitatively evaluate the Painting Assistor, we conducted a user study with 30 participants who freely edited images using our system. Participants rated the Painting

Table 2. Performance comparison between our Painting Assistor and other MLLMs, demonstrating superior visual and semantic consistency in predictions.

Method	GPT-4 [1] Similarity	BERT [10] Similarity	CLIP [49] Similarity
LLaVA-1.5	1.894	0.721	0.795
LLaVA-Next	1.941	0.716	0.794
GPT-4o	1.976	0.684	0.790
Ours	**2.712**	**0.749**	**0.824**

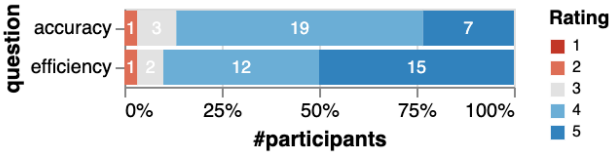


Figure 8. User ratings for the Painting Assistor, focusing on its prediction accuracy and efficiency enhancement capabilities.

Assistor on a 5-point scale for prediction accuracy (1: very poor, 5: excellent) and efficiency facilitation (1: significantly reduced, 5: significantly enhanced). As shown in Fig. 8, 86.67% of users rated prediction accuracy at least 4, validating the ability of our fine-tuned MLLM to interpret user intentions. Similarly, 90% rated efficiency facilitation 4 or above, confirming that Draw&Guess effectively streamlines the editing process by reducing manual prompt inputs. The average scores for accuracy and efficiency were 4.07 and 4.37. We further provide a quantitative analysis with 10 users performing 10 edits, showing an average time savings of 24.92% on iPad per edit and 19.58% on PC per edit, as in the Tab. 3.

Table 3. Editing Time Comparison w./w.o. Painting Assistor.

iPad		PC	
w. Paint. Assit.	w.o. Paint. Assit.	w. Paint. Assit.	w.o. Paint. Assit.
13.29s	17.70s (+4.41s)	12.49s	15.53s (+3.04s)

4.3. Idea Collection Effectiveness and Efficiency

Collecting user ideas effectively and efficiently is critical for the usability and adoption of interactive systems, especially in creative applications where user engagement is crucial. To evaluate the Idea Collector, we conducted a user study with 30 participants, comparing our system against a baseline system on the following dimensions:

- *Complexity and Efficiency* measures how streamlined and intuitive the user finds the system for creative editing.
- *Consistency and Integration* assesses whether the system maintains a cohesive interface and interaction design.
- *Ease of Use* captures the learnability of the system, especially for users with varying backgrounds.
- *Overall Satisfaction* reflects users' general satisfaction with the design, features, and usability of the system.

Baseline: The baseline system was implemented as a customized ComfyUI workflow, replacing our Idea Collector interface with an open-source canvas, Painter Node [46]. This setup enables the focus on the value provided with our Idea Collector by controlling other variables.

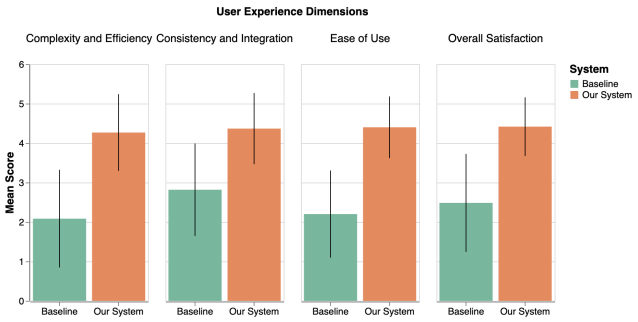


Figure 9. Comparative user ratings between our system and the baseline, with standard deviation shown as error bars.

Procedure: The study lasted approximately 30 minutes for each participant with two systems (our system and the baseline). Each session began with a brief introduction to the system using the case illustrated in Fig. 1. Participants then had 5 minutes to freely explore and edit images. After using both systems, participants completed a questionnaire with 22 questions (10 questions per system covering all four dimensions and 2 questions regarding the Painting Assistor detailed in Sec. 4.2). We employed the System Usability Scale (SUS) [6] for scoring, using a Likert scale from 1 (strongly disagree) to 5 (strongly agree), to capture a global view of subjective usability for each system.

As shown in Fig. 9, our system demonstrated significantly higher scores across all dimensions compared to the baseline. Indicating the effectiveness of our Idea Collector. Further details can be found in the supplementary.

5. Conclusion

In conclusion, our interactive image editing system `MagicQuill` effectively addresses the challenges of performing precise and efficient edits by combining the strengths of the Editing Processor, Painting Assistor, and Idea Collector. Our comprehensive evaluations demonstrate significant improvements over existing methods in terms of controllable generation quality, editing intent prediction accuracy, and user interface efficiency. For future work, we aim to expand the capabilities of our system by incorporating additional editing types, such as reference-based editing, which would allow users to guide modifications using external images. We also plan to implement layered image generation to provide better editing flexibility and support for complex compositions. Moreover, enhancing typography support will enable more robust manipulation of textual elements within images. These developments will further enrich our framework, offering users a more versatile and powerful tool for creative expression. The system is available at https://magic-quill.github.io.

Acknowledgments. This work was supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant number 16212623 and the Ant Group Research Intern Program.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7, 8

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2015. 5

[3] Qingyan Bai, Hao Ouyang, Yinghao Xu, Qiuyu Wang, Ceyuan Yang, Ka Leong Cheng, Yujun Shen, and Qifeng Chen. Edicho: Consistent image editing in the wild. *arXiv preprint arXiv:2412.21079*, 2024. 2

[4] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[5] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. 3

[6] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 1996. 8

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[8] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023. 2

[9] Ka Leong Cheng, Qiuyu Wang, Zifan Shi, Kecheng Zheng, Yinghao Xu, Hao Ouyang, Qifeng Chen, and Yujun Shen. Learning naturally aggregated appearance for efficient 3d editing. In *Proceedings of the International Conference on 3D Vision*, 2025. 2

[10] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 7, 8

[11] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 2

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 2023. 1, 2

[13] Kunyu Feng, Yue Ma, Bingyuan Wang, Chenyang Qi, Haozhe Chen, Qifeng Chen, and Zeyu Wang. Dit4edit: Diffusion transformer for image editing. *arXiv preprint arXiv:2411.03286*, 2024. 2

[14] Yingchaojie Feng, Xingbo Wang, Kam Kwai Wong, Sijia Wang, Yuhong Lu, Minfeng Zhu, Baicheng Wang, and Wei Chen. Promptmagician: Interactive prompt engineering for text-to-image creation. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3

[15] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 2

[16] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 2

[17] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 2, 7

[19] Yingqing He, Zhaoyang Liu, Jingye Chen, Zeyue Tian, Hongyu Liu, Xiaowei Chi, Runtao Liu, Ruibin Yuan, Yazhou Xing, Wenhai Wang, et al. Llms meet multimodal generation and editing: A survey. *arXiv preprint arXiv:2405.19334*, 2024. 2

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020. 2

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[22] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525*, 2024. 1, 2

[23] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 6, 7

[24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2, 7

[25] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019. 2

[26] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image

inpainting model with decomposed dual-branch diffusion. *arXiv preprint arXiv:2403.06976*, 2024. 1, 2, 3, 4, 5, 6, 7

[27] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. *International Conference on Learning Representations*, 2024. 2

[28] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 2

[29] Kangyeol Kim, Sunghyun Park, Junsoo Lee, and Jaegul Choo. Reference-based image composition with sketch via structure-aware diffusion model. *arXiv preprint arXiv:2304.09748*, 2023. 2

[30] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[31] Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. Large-scale text-to-image generation models for visual artists' creative works. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2023. 3

[32] Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 2024. 2

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[34] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2, 6, 7

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2024. 2, 4, 5, 7

[36] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. *arXiv preprint arXiv:2303.05125*, 2023. 2

[37] Zhiheng Liu, Yifei Zhang, Yujun Shen, Kecheng Zheng, Kai Zhu, Ruili Feng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones 2: Customizable image synthesis with multiple subjects. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. 2

[38] Zhiheng Liu, Ka Leong Cheng, Xi Chen, Jie Xiao, Hao Ouyang, Kai Zhu, Yu Liu, Yujun Shen, Qifeng Chen, and Ping Luo. Manganinja: Line art colorization with precise reference following. *arXiv preprint arXiv:2501.08332*, 2025. 2

[39] Weihang Mao, Bo Han, and Zihao Wang. Sketchffusion: Sketch-guided image editing with diffusion model. In *2023 IEEE International Conference on Image Processing (ICIP)*, 2023. 2

[40] Naoki Matsunaga, Masato Ishii, Akio Hayakawa, Kenji Suzuki, and Takuya Narihira. Fine-grained image editing by pixel-wise guidance using diffusion models. *arXiv preprint arXiv:2212.02024*, 2022. 1, 2

[41] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 2, 7

[42] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 1, 2

[43] Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: Sde beats ode in general diffusion-based image editing. *arXiv preprint arXiv:2311.01410*, 2023.

[44] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 1, 2

[45] Xiaohan Peng, Janin Koch, and Wendy E. Mackay. Designprompt: Using multimodal interaction for design exploration with generative ai. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, 2024. 3

[46] Aleksey Petrov. Comfyui custom nodes alekpet. https://github.com/AlekPet/ComfyUI_Custom_Nodes_AlekPet, 2024. 8

[47] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 5

[48] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018. 2

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 2021. 7, 8

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 2, 3

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 2015. 4

[52] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[53] Jaskirat Singh, Jianming Zhang, Qing Liu, Cameron Smith, Zhe Lin, and Liang Zheng. Smartmask: Context aware high-

fidelity mask generation for fine-grained object insertion and layout control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[55] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. Imprint: Generative object compositing by learning identity-preserving representation. *arXiv preprint arXiv:2403.10701*, 2024. 2

[56] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 5

[57] Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd van Steenkiste, Ranjay Krishna, et al. Dreamsync: Aligning text-to-image generation with image understanding feedback. In *Synthetic Data for Computer Vision Workshop@ CVPR 2024*, 2023. 2

[58] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 2

[59] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[60] Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A picture is worth more than 77 text tokens: Evaluating clip-style models on dense captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5

[61] Jue Wang, Yuxiang Lin, Tianshuo Yuan, Zhi-Qi Cheng, Xiaolong Wang, Jiao GH, Wei Chen, and Xiaojiang Peng. Flexedit: Marrying free-shape masks to vllm for flexible image editing. *arXiv preprint arXiv:2408.12429*, 2024. 2

[62] Tengfei Wang, Hao Ouyang, and Qifeng Chen. Image inpainting with external-internal learning and monochromic bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[63] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024. 3

[64] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *arXiv preprint arXiv:2407.05600*, 2024. 2

[65] Chufeng Xiao and Hongbo Fu. Customsketching: Sketch concept extraction for sketch-based image synthesis and editing. *arXiv preprint arXiv:2402.17624*, 2024. 2

[66] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024. 2

[67] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[68] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*, 2024. 2

[69] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, 2020. 2

[70] Shiyuan Yang, Xiaodong Chen, and Jing Liao. Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023. 2, 7

[71] Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, Hideki Koike, et al. Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation. *Advances in Neural Information Processing Systems*, 2024. 1, 2

[72] 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. 2

[73] Yu Zeng, Zhe Lin, and Vishal M Patel. Sketchedit: Mask-free local image manipulation with partial sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 2, 6, 7

[74] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 2024. 2

[75] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3, 4, 5, 6, 7

[76] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 7

[77] Xin Zhang, Jiaxian Guo, Paul Yoo, Yutaka Matsuo, and Yusuke Iwasawa. Paste, inpaint and harmonize via denoising: Subject-driven image editing with pre-trained diffusion model. *arXiv preprint arXiv:2306.07596*, 2023. 1, 2

[78] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *arXiv preprint arXiv:2312.03594*, 2023. 1, 2