DoodleAssist: Progressive Interactive Line Art Generation With Latent Distribution Alignment

Haoran Mo, Yulin Shen, Edgar Simo-Serra, and Zeyu Wang

***Abstract*—Creating high-quality line art in a fast and controlled manner plays a crucial role in anime production and concept design. We present *DoodleAssist*, an interactive and progressive line art generation system controlled by sketches and prompts, which helps both experts and novices concretize their design intentions or explore possibilities. Built upon a controllable diffusion model, our system performs progressive generation based on the last generated line art, synthesizing regions corresponding to drawn or modified strokes while keeping the remaining ones unchanged. To facilitate this process, we propose a latent distribution alignment mechanism to enhance the transition between the two regions and allow seamless blending, thereby alleviating issues of region incoherence and line discontinuity. Finally, we also build a user interface that allows the convenient creation of line art through interactive sketching and prompts. Qualitative and quantitative comparisons against existing approaches and an in-depth user study demonstrate the effectiveness and usability of our system. Our system can benefit various applications such as anime concept design, drawing assistant, and creativity support for children.**

***Index Terms*—Progressive line art generation, sketch-guided generation, iterative system, regional latent blending, controllable diffusion models.**

I. Introduction

CREATING high-quality line art in an efficient and controllable way plays a fundamental role in anime production, concept design, and illustration creation. Artists typically follow a progressive procedure to draw each part step by step in their creation workflows [4], [5], as this can provide immediate feedback to guide the next action [6], [7]. Additionally, iterative refinement is typical when artists intend to adjust local regions while keeping satisfying parts unchanged.

While existing diffusion model-based tools have demonstrated potential in assisting the creation process with sketches as input [3], [8], they struggle with progressive line art generation. Block-and-Detail [1] is designed for iterative sketch-to-image generation for refining the outputs, but, as a generic image generator, it tends to produce undesired images with a realistic style or line art drawn on paper, as shown in Fig. 1 (top). In addition, it fails to maintain consistency with input sketches (e.g., the extra balloons) or the last generated image (e.g., the face of the boy), indicating its weakness in region control during the progressive generation process. As another solution, the mask-guided completion method that combines an inpainting model [2] and a sketch-oriented ControlNet [3] fails to ensure smooth transition and style consistency with the input partial line art, as shown in Fig. 1 (bottom).

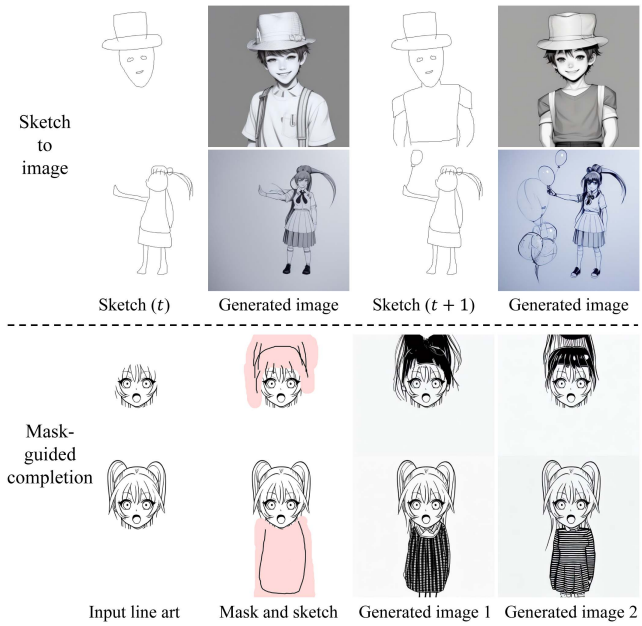


Fig. 1. Limitations of existing diffusion model-based methods on progressive line art generation. The top ones are from an iterative sketch-to-image generation system [1], and the bottom ones from a Stable Diffusion Inpainting model [2] combined with sketch-oriented ControlNet [3]. Note that trigger words such as "line art," "monochrome," and "white background" are added to the prompts.

To address the issues above, we introduce *DoodleAssist*, an interactive system for high-quality line art generation in a progressive manner guided by sketches and prompts. As shown in Fig. 2(a), it supports a step-by-step creation process with region control based on the last generated result, which updates regions corresponding to newly drawn or modified strokes and keeps satisfying ones unchanged. As shown in Fig. 2(b), the system works on different styles of sketches for diverse generation, including single characters, animals, objects, multiple characters,

Received 18 July 2025; revised 10 October 2025; accepted 14 October 2025. Date of publication 23 October 2025; date of current version 6 February 2026. This work was supported in part by the National Natural Science Foundation of China under Project 62502410 and in part by Guangzhou Industrial Information and Intelligent Key Laboratory under Project 2024A03J0628. Recommended for acceptance by C. Li. *(Corresponding author: Zeyu Wang.)*

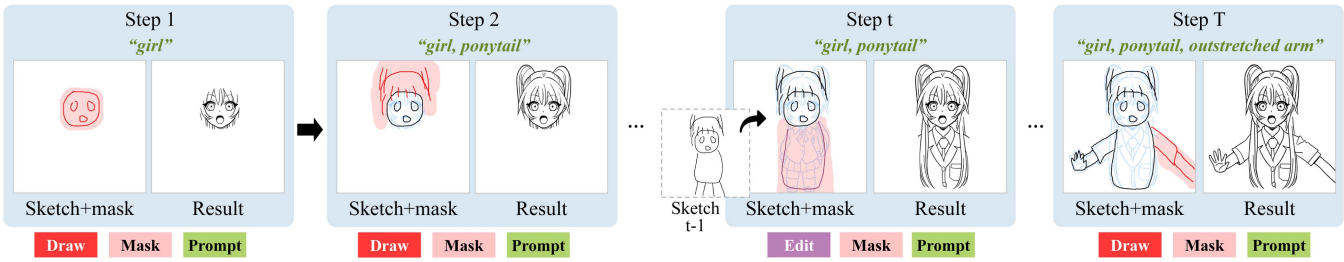
Haoran Mo and Yulin Shen are with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China.

Edgar Simo-Serra is with Waseda University, Tokyo 169-8050, Japan.

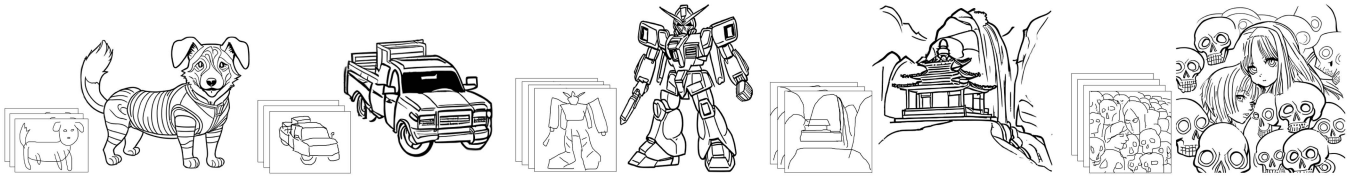
Zeyu Wang is with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the Hong Kong University of Science and Technology, Hong Kong (e-mail: zeyuwang@ust.hk).

Our code is available at https://github.com/MarkMoHR/DoodleAssist.

Digital Object Identifier 10.1109/TVCG.2025.3624800



(a) Workflow of our interactive system



(b) Diverse generation

Fig. 2. Workflow of our interactive and progressive line art generation system *DoodleAssist*, and its performance on diverse generation. (a) The progressive process generates or updates designated regions (in pink) step by step according to drawn or modified strokes in the sketches, while ensuring smooth transition with those unchanged regions. The actions at each step are shown with color coding. (b) The system can be applied to synthesize diverse line art images, such as single characters, animals, objects, multiple characters, and complex scenes.

and complex scenes. The system applies to both novice users and experts. For novices with little drawing experience, it helps them create line art conveniently and improve their drawing skills. For expert users, the system can quickly concretize their design intentions and provide them with inspiration.

To facilitate a smooth transition between mask regions and the remaining ones in the progressive generation process, we propose a *latent distribution alignment mechanism* that improves seamlessness of regional latent blending in diffusion models. Directly blending two regions in the latent space tends to induce incoherence of content and style [9] and line discontinuity in our line art-specific task. Existing methods for addressing this issue [10], [11], [12] are either slow or ineffective in this task. We find that an effective and efficient way to ensure transition between regions of line art is to align their data distributions in the latent space. Thus, we propose a plug-and-play solution through statistical matching. Our mechanism helps to improve the blending of the regions without introducing additional runtime overhead.

To enable users to conveniently create line art through sketches and prompts, we built an interactive user interface for our progressive line art generation system, which provides stroke, mask, and prompt tools, as shown in Fig. 2(a). We evaluated the performance of *DoodleAssist* and the proposed mechanism through qualitative and quantitative comparisons against existing methods. We also conducted an in-depth user study with eight novices and eight experts to validate the system's effectiveness and usability.

The main contributions of this work are as follows:

- An interactive and progressive line art generation system *DoodleAssist* controlled by user-drawn sketches, which updates intended regions while preserving remaining ones. It helps to concretize users' intentions step by step in their creation processes.
- A latent distribution alignment mechanism to facilitate region control, which improves the transition between regions during latent blending and alleviates regional incoherence and line discontinuity.
- In-depth evaluation of the system through comparisons against existing approaches and a user study with novice and expert users.

II. Related Work

A. Sketching and Drawing Systems

Sketch serves as a convenient and intuitive tool to express ideas and convey concepts quickly, and thus is widely used in content creation [1], [13], [14], [15], [16], drawing assistant [6], [17], [18], 3D prototyping with AR/VR [19], [20], [21], animation production [22], [23], [24], etc. Among these systems, Block-and-Detail [1] is closest to our framework, which supports an iterative refinement process to create images with freehand sketches. As a generic image generator, it struggles with high-quality line art generation, unlike our line art-specific system. Besides, ours differs noticeably from it in design concepts. First, it produces complete images from partial sketches to provide scaffolding for users, which brings about unnecessary disturbance in line art creation. Therefore, our system synthesizes contents within regions of updated strokes to enable a progressive generation process with iterative sketching. Second, intermediate results in the Block-and-Detail system only serve as feedback to help users update the sketch to obtain the final image, and are discarded then. On the contrary, in our step-by-step line art creation, previously generated parts that users are satisfied with remain unchanged and make up the final image.

B. Image Generation With Diffusion Models

Diffusion models [25], [26], [27] have become the mainstream approach for image generation, due to the powerful generative ability derived from large-scale pretraining [2]. Controllable diffusion models allow users to direct the diffusion process [3], [8]

through an additional condition. While quite a few works [3], [8], [28], [29], [30], [31] have focused on sketch-guided generation of objects, human portraits, and scenes, controllably producing high-quality line art remains rarely explored. Moreover, they have difficulty in progressive generation, with all pixels changing during each denoising step. In contrast, our system supports a progressively regional generation process aligning with users' production workflow, in which newly generated regions are coherently blended with the preserved ones due to the proposed latent distribution alignment mechanism.

C. Line Art Generation

Our task synthesizes fine-grained line art from sparse sketches. An early work called SketchMan [32] starts to study this creative generation task and names it "sketch enhancement." Built upon conditional generative adversarial networks (GANs) [33], it exhibits poor line art results with undesired artifacts such as blurry lines and discontinuous strokes. By comparison, our method produces much higher-quality line art while allowing a progressive generation process.

AniFaceDrawing [34] explores generating line art based on interactive sketching, although limited to synthesizing anime portraits. Built upon a GAN inversion technique, it tends to change original regions when adding new strokes, violating the progressive generation rule. In contrast, our system can create diverse line art images apart from the portraits. Moreover, it enables the preservation of satisfying regions through the progressive generation process.

D. Regional Blending in Diffusion Models

Regional blending is broadly used in diffusion model-driven tasks like inpainting [10], [12], [35], [36] and text-based image editing [9], [11], [37], [38], [39]. During the diffusion reverse process, they replace predicted noisy latent in known regions with a noisy version of a given image. Given that directly blending two parts in either pixel or latent space tends to induce incoherence [9], [39] or disharmonious boundaries [10], several strategies are introduced to address these issues. Blended Latent Diffusion [39] proposes a per-image fine-tuning method for the VAE decoder, although it requires much longer inference time and does not apply to general models. SDEdit [11] and RePaint [10] introduce resampling strategies by repeating the full-step reverse process several times [11] or the sampling at each reverse step [10]. Increased runtime is introduced due to the additional sampling steps. SmartBrush [12] uses clean background features without adding noise for the blending to provide rich global context when fine-tuning the diffusion models. It fails in our task where the pretrained weights should remain unchanged, due to the conflict between predicted noise and clean background features. To address the blending issue without introducing additional inference time overhead, we propose a simple yet effective and efficient mechanism through latent distribution alignment in the regional latent blending process. It offers a performance boost for the progressive generation by improving the regional transition.

III. The *DoodleAssist* System

A. Overview

We develop *DoodleAssist*, a sketch-guided progressive line art generation system, which is intended to meet two main requirements of users in their interactive creation process:

- *R1:* Regional control based on partial sketches.
- *R2:* Preservation of satisfying parts and seamless blending with newly generated ones.

The requirements are found from a formative user study with a baseline method [1]. Please refer to the supplementary material for more details and discussion.

As illustrated in Fig. 3(a), the system produces line art step by step by updating regions of the incoming or modified strokes (**R1**), while keeping satisfying parts from previous generations unchanged (**R2**). The system is built by applying ControlNeXt [29] iteratively, a trainable controllable diffusion model with an input sketch c_s as a condition. At each iteration, the encoded sketch features along with a prompt c_p are integrated into a frozen latent diffusion model (LDM) [2].

We introduce a progressive generation process with region control, which is enabled by regional latent blending. As illustrated in Fig. 3(b), the noise prediction and denoising procedures account for a designated region (e.g., the added hair) indicated by a mask M, and the remaining one $(1 - M)$ is replaced with a noisy latent of known pixels from the last generated line art.

Seamless blending of two noisy latents is essential, but a direct blending could not guarantee the coherence [9], [39]. We propose a *latent distribution alignment mechanism* (Fig. 3(c)) that improves transitions between two regions in the latent space, which helps to alleviate artifacts in the synthesized line art, such as incoherence of content and style and line discontinuity.

To enable users to create line art by sketching step by step using our proposed algorithm, we build a user interface for our progressive line art generation system *DoodleAssist*, which incorporates stroke and mask tools.

B. Progressive Generation With Regional Latent Blending

To overcome the issue of existing iterative sketch-to-image generation approaches [1] that make unexpected changes to unedited regions, as shown in Fig. 1, we introduce a progressive generation process with region control. As shown in Fig. 3(a), it updates regions of the line art step by step according to the iterative sketches.

The key to the progressive process is regional latent blending in the latent diffusion model. As illustrated in Fig. 3(b), in each denoising step, with an input raw latent denoted as $\tilde{x}_t$ and a mask M for the intended region to be changed, such as the hair, we calculate a masked latent $\tilde{x}_t \odot M$ ($\odot$ denotes element-wise product). Given that the remaining regions, e.g., the face and the background, are already known from the last generated line art b, we directly add random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to b for a noisy version $\tilde{b}_t$, and use the inverse mask $1 - M$ to obtain the corresponding regions $\tilde{b}_t \odot (1 - M)$. Afterwards, the modified latent x_t is formulated as a blend of the two masked noisy latents:

$$x_t = \tilde{x}_t \odot M + \tilde{b}_t \odot (1 - M), \tag{1}$$

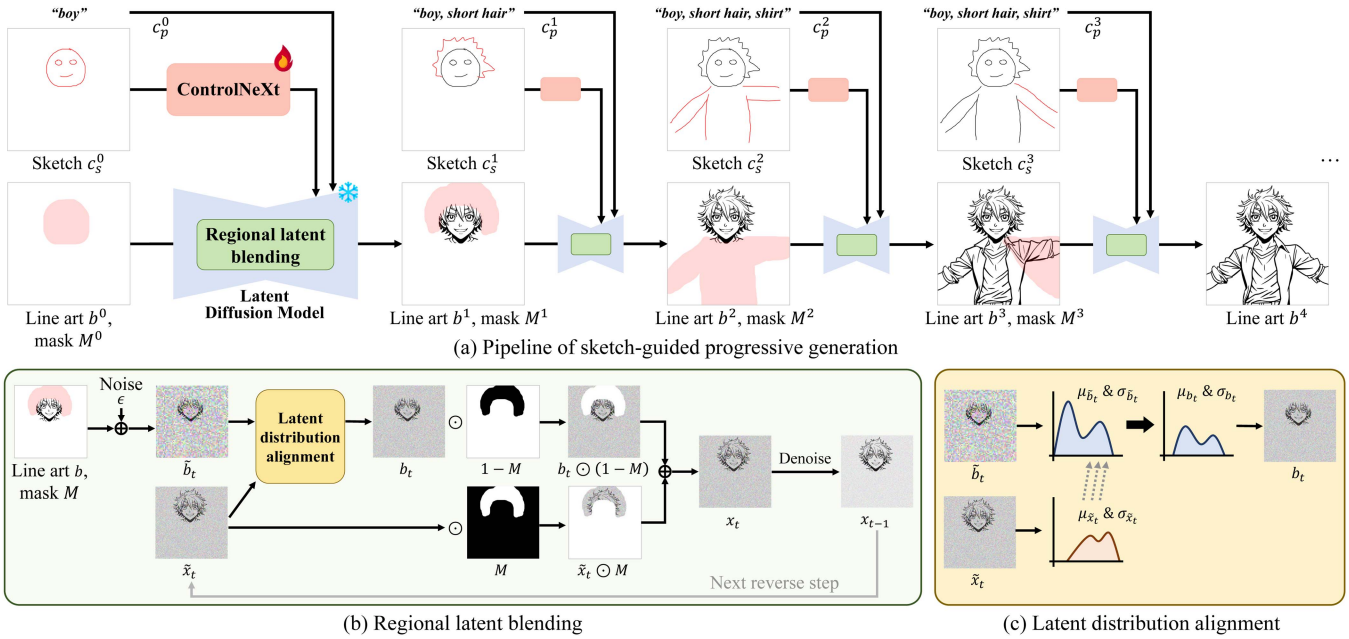


Fig. 3. Framework of our progressive line art generation system *DoodleAssist*. (a) It is built by applying a controllable diffusion model iteratively to update intended regions for incoming or modified strokes. (b) Regional latent blending enables the progressive process with region control, which accounts for noise prediction and denoising in the intended region M. The remaining region $1-M$ is replaced with a known latent from the last generated line art. (c) A latent distribution alignment mechanism is proposed to improve transitions between the two regions. Newly added or modified strokes are highlighted in red. Masks in pink indicate the intended regions.

which is then used for the denoising procedure.

As the unmasked regions are derived from known pixels, we enforce a *regional supervision* for only the masked regions during training. Specifically, the regression loss is calculated only in the mask regions M with 1 corresponding to the intended area and 0 otherwise, which is formulated as:

$$\mathcal{L} = \mathbb{E}_{x_0,t,c_p,c_s,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|M * (\epsilon - \epsilon_\theta(x_t, t, c_p, c_s))\|^2\right]. \tag{2}$$

The regional loss helps the model concentrate on the mask regions, boosting its performance in terms of alignment with the sketches and quality of the line art.

C. Latent Distribution Alignment Mechanism

The regional generation process requires blending two regions in latent space. Still, prior studies have found that the result of directly blending two noisy images is not guaranteed to be coherent [9], [39], and will result in artifacts such as disharmonious boundaries [10], [12]. As shown in Fig. 4, when blending naively ("w/o latent alignment"), the produced line art images exhibit incoherence of content and style between newly generated regions and original ones. What's worse is that the discontinuity of lines appears.

Several works have attempted to overcome this issue by fine-tuning the VAE decoder on a per-image basis [39] or adopting resampling strategies [10], [11]. However, they are either non-general or slow in inference. In contrast, we are interested in a solution that applies to general models while avoiding additional runtime overhead.

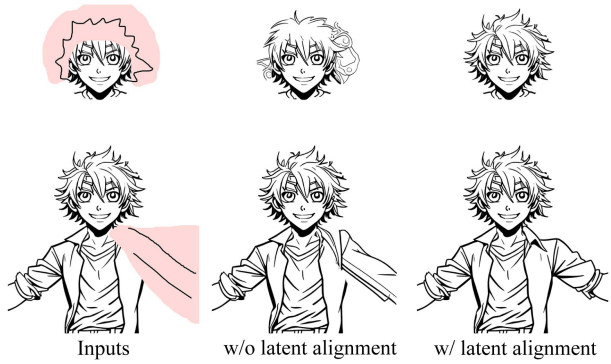
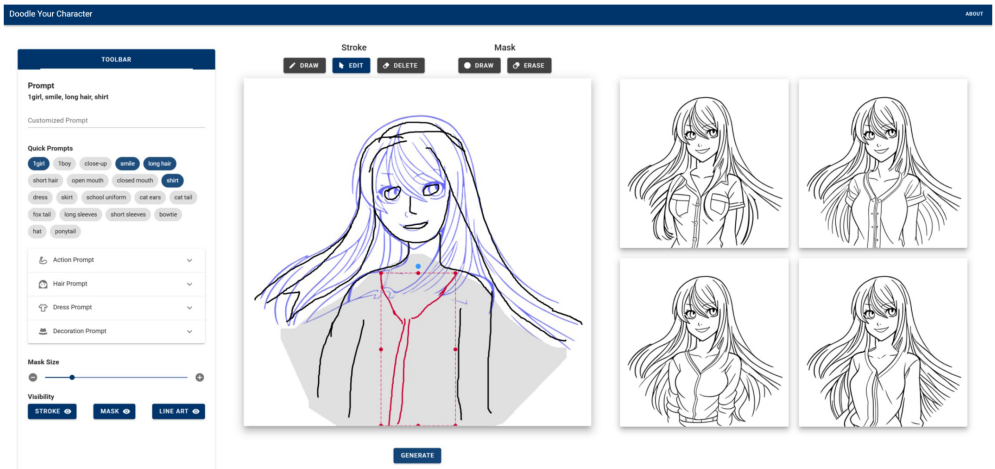


Fig. 4. Motivation of latent distribution alignment mechanism. Results without the latent alignment exhibit line discontinuity and inconsistent content and style between newly generated and original regions. Masks in pink indicate the edited parts.

Inspired by a work [40] that observed two latent spaces typically exhibit inconsistent data distributions, we propose a *latent distribution alignment mechanism* to align two latent spaces before blending, which is intended to improve transitions between them and internal compatibility inside the blended result. The core of this mechanism is to match the distributions of two latents statistically through normalization and rescaling. Specifically, as shown in Fig. 3(c), the mean and variance of the predicted denoised latent $\tilde{x}_t$ are exploited to normalize the noisy latent $\tilde{b}_t$ derived from the last generated line art b. With the calculated mean-variance pairs $(\mu_{\tilde{x}_t}, \sigma_{\tilde{x}_t})$ and $(\mu_{\tilde{b}_t}, \sigma_{\tilde{b}_t})$, a normalized noisy latent is formulated as:

$$b_t = \left(\tilde{b}_t - \mu_{\tilde{b}_t}\right) \div \sigma_{\tilde{b}_t} \times \sigma_{\tilde{x}_t} + \mu_{\tilde{x}_t}. \tag{3}$$



(a) Tool Panel (b) Canvas Panel (c) Output Panel

Fig. 5. User interface of *DoodleAssist*. The gray patch in (b) is the mask. The strokes in red are those to be edited. Please refer to the supplemental video for the interactive usage of this interface.

Then, the latent alignment is reformulated according to (1):

$$x_t = \tilde{x}_t \odot M + b_t \odot (1 - M)\,. \tag{4}$$

The latent distribution alignment mechanism is plug-and-play, and we incorporate it into each regional latent blending step. It is simple yet effective, bringing a noticeable performance boost to the synthesized images with smooth transition and coherent content between the edited and original regions, as shown in Fig. 4. The experiments also show that the mechanism is efficient without introducing additional inference time overhead.

D. System Interface

We built a web interface for our interactive line art generation system *DoodleAssist*. As shown in Fig. 5, it contains a tool panel, a canvas panel, and an output panel. In the canvas panel (b), users can first select a stroke tool, including drawing, editing through transformations (i.e., translation, rotation, and scaling), or deleting. The system automatically computes a mask suggesting changing regions for each stroke operation. Users can edit the suggested mask by drawing or erasing regions using the tools at the top-right corner in (b). The brush size of the mask tools can be adjusted using the "Mask Size" slider in the tool panel (a). Furthermore, the canvas panel uses an overlay to show the last generated line art, which guides users to take the following actions. The visibility of each layer can be toggled at the bottom of the tool panel.

Before clicking the "GENERATE" button at the bottom, users can input a prompt in the tool panel (a) by selecting quick prompts or typing in words. The quick prompts are those tags statistically prominent in our line art dataset for anime characters. After the generation, the output panel (c) displays four results, and users can select one, which will be updated to the canvas panel immediately.

The system is built on a desktop with a mouse for drawing. The interface is built with a web front-end and a GPU-based back-end. The front-end is implemented with Vue. The back-end is deployed on a server with four NVIDIA GeForce RTX 4090 GPUs, each processing a single image in parallel to speed up the interaction. Each generation practice with the interface takes 1.6 seconds on average.

IV. Experiments

A. Dataset and Implementation Details

1) Dataset: We rely on the SketchMan dataset [32], which contains pairs of freehand sketches and corresponding line art images, to train our model. The dataset includes 2,120 pairs for training and 127 for validation. As we require progressive sketches and line art, we vectorize the sketch images into separated strokes [41], and then apply a rule-based grouping algorithm to synthesize progressive sketches. The algorithm mimics the sketching process by randomly choosing a starting position, and then progressively finding the next strokes for adding into the group according to their distances to the starting position or the lastly grouped stroke (see the supplementary material for more details of the algorithm). Afterwards, we synthesize the progressive line art images by applying the AlphaShape algorithm [42] to the sketches for foreground masks used to crop partial line art. Finally, we obtain 144,826 pairs for training and 9,975 for validation, including progressive sketches and line art images of varying completion degrees in resolution 512×512. Fig. 6 shows some examples. Although the sketches are collected by adding strokes only, the trained model generalizes well to cases of editing and deleting strokes. Our approach still relies on textual prompts as input, so we employ a captioning method for anime images [43] to collect the prompt data.

2) Implementation Details: We employ ControlNeXt [29] as the controllable model due to its proven strengths of fewer trainable parameters and faster convergence compared to ControlNet [3]. We choose a Stable Diffusion v1.5 model fine-tuned on line art data from an open-source platform [44] as the backbone diffusion model. We adopt a controlling scale factor 0.9 to inject the condition features from ControlNeXt into the UNet model.

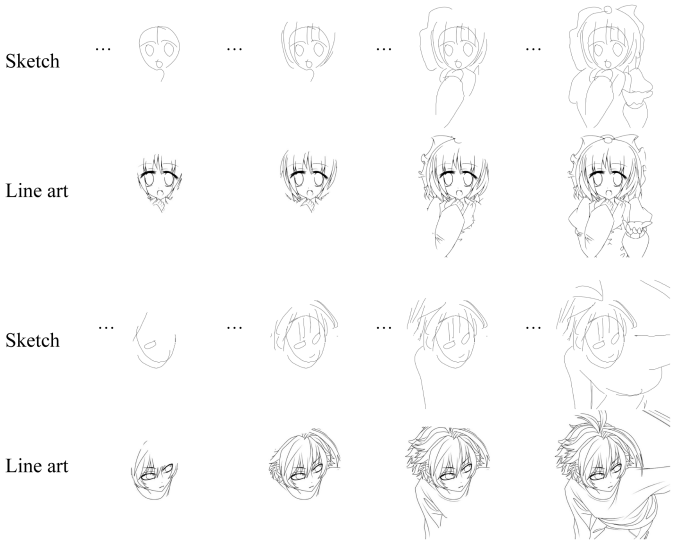


Fig. 6. Examples of training and validation datasets with iterative sketches.

We train the ControlNeXt for two epochs with a batch size of 6. In the inference stage, we use UniPC [45] as the sampling method with 20 sampling steps to speed up the denoising process. The system calculates suggested masks automatically by using the AlphaShape algorithm [42] to create a convex hull for the added, edited, or deleted strokes.

B. Comparison With Existing Approaches

We compare *DoodleAssist* with existing approaches in four settings: Based on complete sketches, based on iterative sketches, mask-guided completion, and a user study to evaluate the practical usefulness.

1) Comparisons Based on Complete Sketches: We compare to SketchMan [32], the first work on sketch to line art generation with a curated dataset of complete sketches and the corresponding line art. It proposes a sketch enhancement framework built on a conditional GAN model [33] along with several training strategies. We chose the best results with the strategy "SS+L1+FM" for comparison.

As shown in Fig. 7(a), SketchMan can generate line art aligned with the sketches, but they exhibit blurry and indistinct lines, insufficient details, and poor visual quality. It is due to the unstable training with a GAN model on the original small-scale paired dataset. In contrast, our approach, built upon the powerful generative prior of a line art-pretrained diffusion model, shows results with precise details, alignment to the sketches, and higher visual quality.

2) Comparisons Based on Iterative Sketches: To the best of our knowledge, progressive sketch to line art generation is rarely explored, except for a work AniFaceDrawing [34]. It generates only anime portraits using iterative sketching, so we use portrait examples in the paper for comparison. We also train ControlNet [3] and original ControlNeXt [29] using our dataset with iterative sketches and the corresponding line art, and conduct a quantitative evaluation with the validation set in two aspects: 1) spatial alignment between the generated line art and the sketches, and 2) visual quality. For the spatial alignment, we adopt the commonly used Chamfer distance [46], [47] and CLIP distance [48], [49] as the metrics. Regarding the visual quality, we use LPIPS [50] and FID [51]. AniFaceDrawing is excluded from the quantitative comparison as its source code has not yet been released.

TABLE I
QUANTITATIVE COMPARISONS WITH LINE ART GENERATION METHODS

	Spatial Alignment		Visual Quality	
	Cham. dist.(↓) (e2)	CLIP dist.(↓)	LPIPS(↓) (e-2)	FID (↓) (e-1)
ControlNet [3]	54.22	11.23	38.76	11.28
ControlNeXt [29]	32.20	5.61	29.75	10.99
Ours	**3.16**	**4.60**	**24.29**	**10.54**

As shown in Fig. 7(b), AniFaceDrawing [34] differs from our approach in generating complete line art according to each sketch. The synthesized results are not monochromatic. Moreover, alignment to sketches is sometimes worse in contrast to ours, such as face contour/short hair in column b4/b5, due to its employed GAN inversion technique that is weak at aligning with the shape of the sketches. Both ControlNet [3] and ControlNeXt [29] struggle with the progressive generation. Without strictly following the strokes, they generate redundant content for sketches of low completion degrees (columns b1 to b3). What is worse, they change regions that do not belong to the new strokes, e.g., the facial expression (ControlNet) and face contour (ControlNeXt) from column b4 to b7. By comparison, our approach works on sketches of varying completion degrees without producing redundant content due to the introduced progressive generation scheme with region control. In addition, newly generated lines harmonize with previous ones, thanks to the proposed latent distribution alignment mechanism that improves transitions between the regions.

The quantitative results are shown in Table I. Both ControlNet [3] and ControlNeXt [29] perform worse than our method in spatial alignment, probably due to the redundant content generated for the sparse strokes in the early stage of iterative sketching. ControlNet tends to synthesize line art with thick lines as shown in Fig. 7(b), leading to the worst visual quality metrics. Our approach is superior to the two alternatives in visual quality, indicating that it produces line art images that look more like the real ones.

3) Comparisons With a Mask-Guided Completion Method: As our task uses a mask and a sketch as inputs, we compare with a mask-guided completion method designed for this task. It combines a Stable Diffusion Inpainting model [2] with sketch-oriented ControlNet [3]. The input prompt is enhanced with trigger words such as "line art," "monochrome," and "white background."

As shown in Fig. 8, the mask-guided completion method produces content in the masked regions with an inconsistent style with the unedited parts, exhibiting dense lines or black patches. This is probably due to the inpainting backbone model pretrained on generic image data with a dense color distribution.

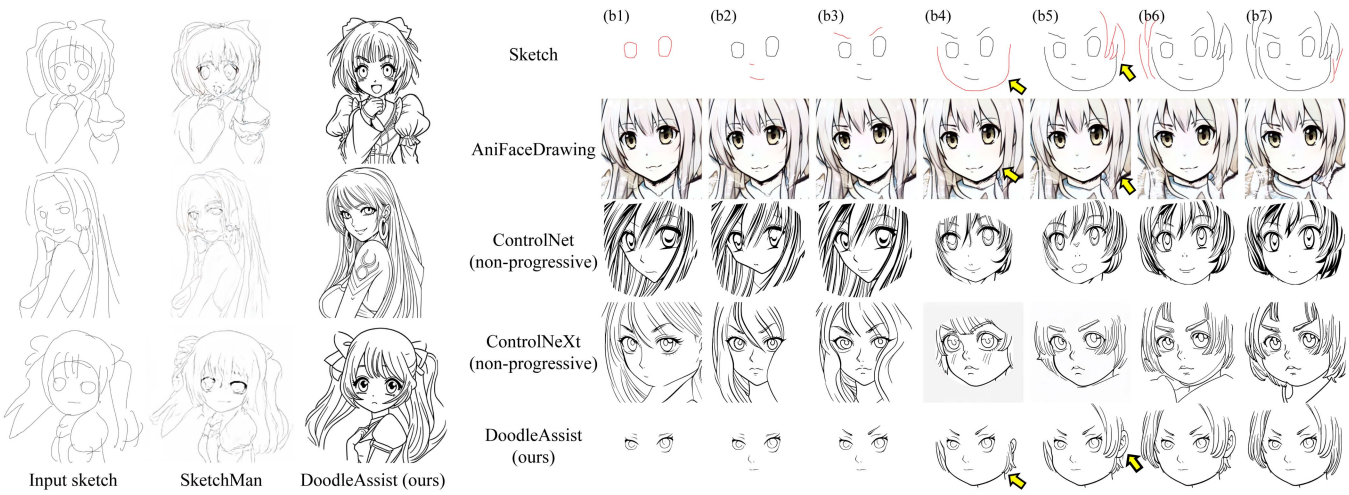


Fig. 7. Comparisons with line art generation methods based on complete or iterative sketches. Sketch examples in (a) are from SketchMan [32], and those in (b) are from AniFaceDrawing [34]. Newly added strokes are highlighted in red. Yellow arrows are used to highlight regions.

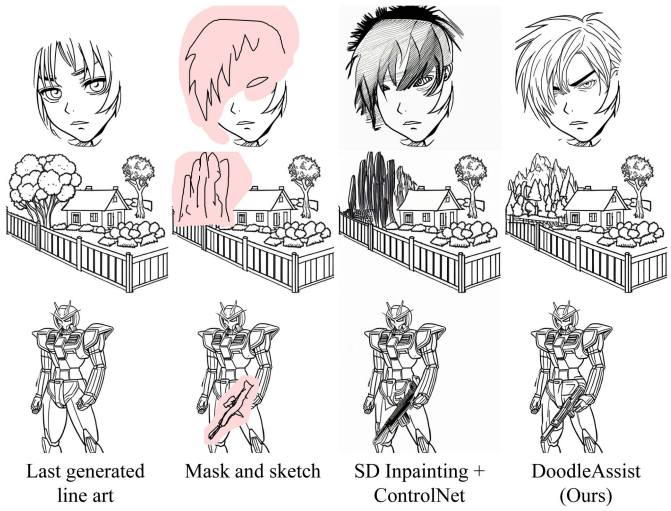


Fig. 8. Comparisons with a mask-guided completion method that combines a Stable Diffusion Inpainting model [2] with sketch-oriented ControlNet [3].

In contrast, our approach generates sparse line drawings that are compatible with the original regions.

4) User Study of Iterative Sketch-to-Image Systems: To evaluate the practical usefulness of progressive line art generation for users, we conducted a user study comparing our system with an existing iterative sketch-to-image generation system named Block-and-Detail [1].

a) Study Design: We recruited 16 users aged between 22 and 32 (8 females and 8 males) for the study. Eight of them are novice users (P1-P8) with no drawing experience. The other eight (P9-P16) are expert users who major in drawing and art, or have formal training in drawing. For a fair comparison, we fine-tuned Block-and-Detail [1] with line art data and deployed the model with a similar interface. Each user was asked to use the baseline system and ours to create line art freely and iteratively. After the experiments, we asked the participants to complete a questionnaire and conducted a structured interview to collect user feedback.

b) Qualitative Results: We summarize the user feedback and distill three key insights, which indicate the usefulness of the progressive line art generation process in our system.

Local adjustment while preserving satisfying regions is preferable: As shown in Fig. 9, P14 drew little girls in both systems for a comparison. After obtaining the initial line art, she edited the local parts such as the hat, the fingers, or the mouth. The resulting line art with Block-and-Detail changed greatly, especially in the facial identity. In contrast, our system modified only the intended local regions while keeping the satisfying regions unchanged. *"The previous system didn't allow me to edit local details. I can't master the generation process,"* said P14, *"The new system helped me too much in the progressive generation."*

Prompts can map to local regions for better controllability: As shown in Fig. 9, P6 drew a cat and a fish in both systems. When adding strokes to the fish and using a prompt "cat ears, cat tail, fish," Block-and-Detail produced a cat with characteristics of the fish, e.g., fish scale and tail (see yellow arrows). While in our system, P6 sketched a fish and used a prompt with the word "fish." The system synthesized an additional fish without interfering with the generated cat. P6 noted, *"In the previous system, I could not adjust the cat or the fish individually. I could not edit a specified region as it changed the entire image. While in the new system, I could make it iteratively with the prompts."*

Regional generation based on partial sketches improves spatial alignment: As demonstrated in Fig. 9, P4 drew a head, and Block-and-Detail produced a complete image misaligned with the partial sketch. By comparison, our system generated highly aligned content within the stroke regions. P4 noted, *"The new system did not generate redundant contents outside the mask region. The results matched better with the sketches."*

c) Quantitative Results: As shown in the questionnaire result in Fig. 10, 81% of users agreed that our system generated aligned results with sketches (Q1), in contrast to 19% for Block-and-Detail [1], indicating that, compared with generating

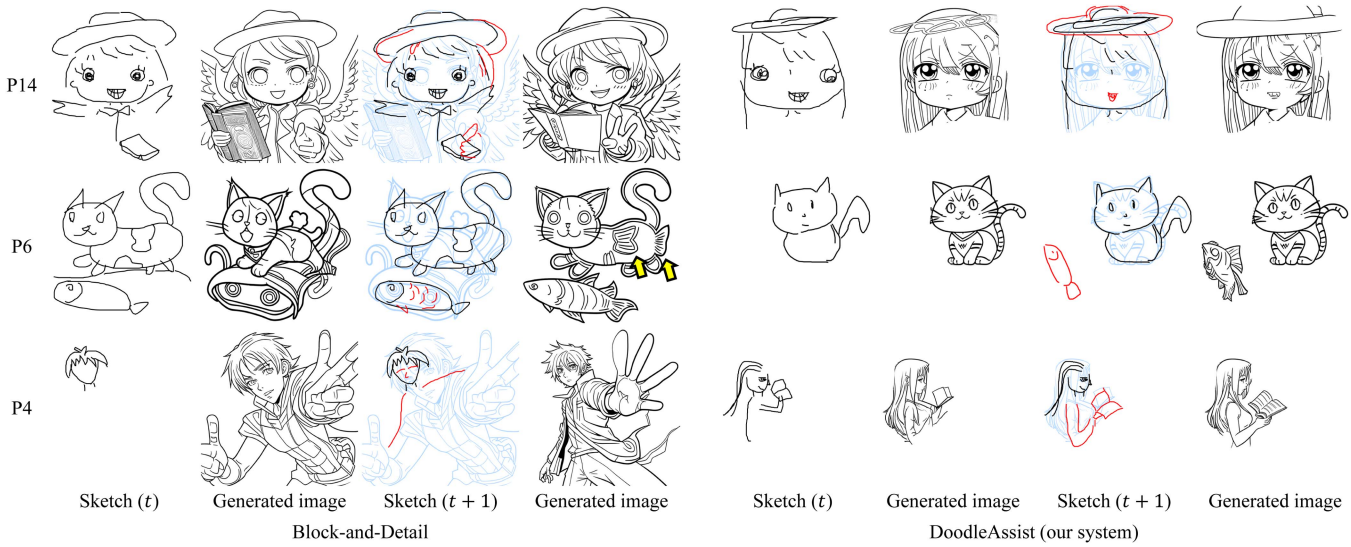


Fig. 9. User study of iterative sketch to line art generation between Block-and-Detail system [1] and ours. Newly added or modified strokes are highlighted in red. Blue drawings underneath are the last generated images. Yellow arrows are used to highlight regions.

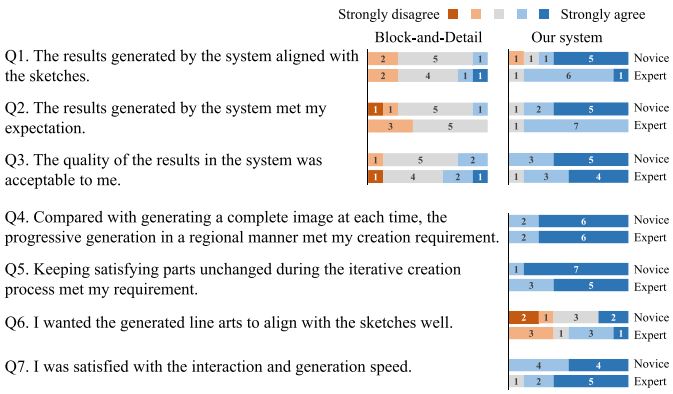


Fig. 10. Ratings of the questionnaire. The numbers in the bar represent the number of participants who submitted the same rating.

complete line art, regional generation within the sketch regions can enhance the spatial consistency. According to Q2 and Q3, at least 87% of users were satisfied with the generated line art in our system, in contrast to at most 31% for Block-and-Detail. This is probably due to the design concepts of our system, since all users, either novices or experts, agreed or strongly agreed that progressive generation in a regional manner and keeping satisfying parts unchanged during the process met their creation requirements (see Q4 and Q5).

C. Effectiveness of Latent Distribution Alignment Mechanism

1) Evaluation Settings: In this experiment, we use sketches with iteratively added strokes as input, and utilize the last generated line art for latent blending. At the first iteration, a blank image is used for the blending. A mask is calculated for the new strokes, which indicates the edited regions and is used in the latent blending. Four latent blending methods are adopted for comparisons. The first is the Blended Latent Diffusion Model (LDM) [39] without latent distribution alignment. The second one is introduced by SmartBrush [12], which directly uses latent features of the last generated line art without adding noise for the blending. The last two are from RePaint [10] and SDEdit [11], where different resampling strategies are employed to improve the blending. For RePaint, we use two resampling steps with a jump length of 2. For SDEdit, we first perform reverse diffusion from $t = T$ to $t = T/2$ and then repeat the reverse process from $t = T/2$ to $t = 0$ three times.

2) Qualitative Comparisons: As shown in Fig. 11, SmartBrush [12] often produces gray patches within the edited regions, indicating an inevitable gap and incompatibility between the predicted noisy latent and the clean latent features from the background. The resampling strategy in RePaint [10] introduces noise into the sketch regions, and the artifact becomes more significant as the number of resampling steps and jump length increase. This is probably because the resampling strategy injects noise back and forth, making the denoising process unstable. The resampling method in SDEdit [11] often makes the generated lines light, probably because its excessive denoising loops treat some stroke pixels as noise and remove them unnecessarily. This also leads to inconsistent line intensities between the original and newly generated regions. Regarding the original latent blending method [39] without distribution alignment, the generated regions exhibit severe incoherence with the previous parts, especially in line continuity. After employing the latent distribution alignment mechanism, which normalizes the distributions for two noisy latents, smooth transitions and line continuity can be observed between the newly generated and original parts. Moreover, a consistent drawing style is ensured. This demonstrates the mechanism's effectiveness, as it helps to improve coherence between the two latent spaces, thus enhancing the performance of the regional blending.

3) Quantitative Comparisons: We surveyed for a quantitative comparison regarding two aspects: 1) spatial alignment between the sketches and the generated line art, and 2) overall quality. We selected 15 examples randomly in our validation set

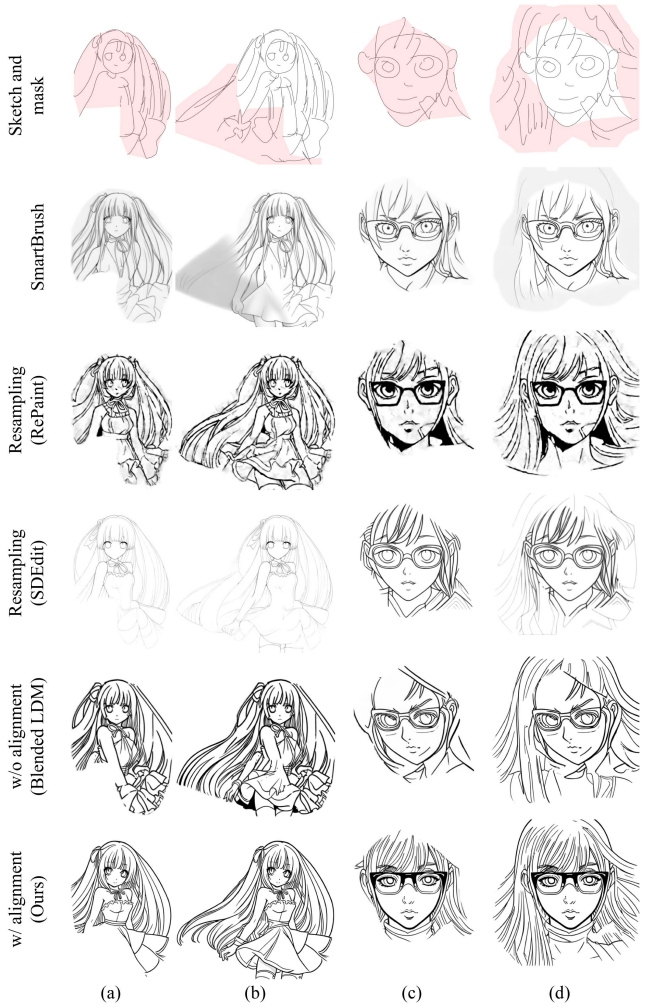


Fig. 11. Qualitative comparisons of latent blending methods. We show two examples with progressive sketches. Masks in pink indicate the edited regions for the newly added strokes and are utilized for the latent blending. In (a) and (c), a blank image is used as the last generated image for blending. In (b) and (d), the generated line art images in (a) and (c) are used for blending.

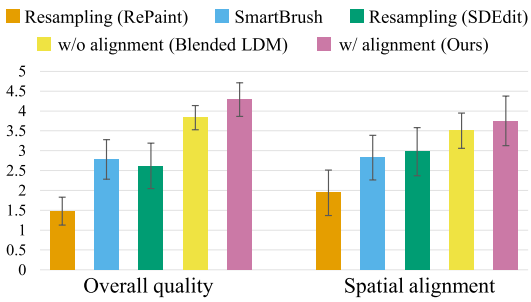


Fig. 12. Quantitative results of latent blending methods through a user survey, where participants ranked results in the two aspects.

and asked participants to rank the generated results from the five methods. 28 users aged between 22 and 35 participated in the survey. As illustrated in Fig. 12, in both measurements, our proposed latent distribution alignment mechanism achieves the best rank, indicating its effectiveness against the alternatives.

TABLE II
RUNTIME COMPARISONS OF LATENT BLENDING METHODS. "SYSTEM INFERENCE" MEANS THE RUNTIME FOR GENERATING FOUR IMAGES SIMULTANEOUSLY USING THE SYSTEM. "BATCH AVERAGE" DENOTES THE AVERAGE RUNTIME OF APPLYING THE ALGORITHM TO 300 EXAMPLES.

Latent Blending Methods	System Inference	Batch Average
Resampling (SDEdit [11])	3.4s	1.40s
Resampling (RePaint [10])	3.1s	1.37s
SmartBrush [12]	1.6s	0.73s
w/o alignment (Blended LDM [39])	1.6s	0.74s
w/ alignment (Ours)	1.6s	0.73s

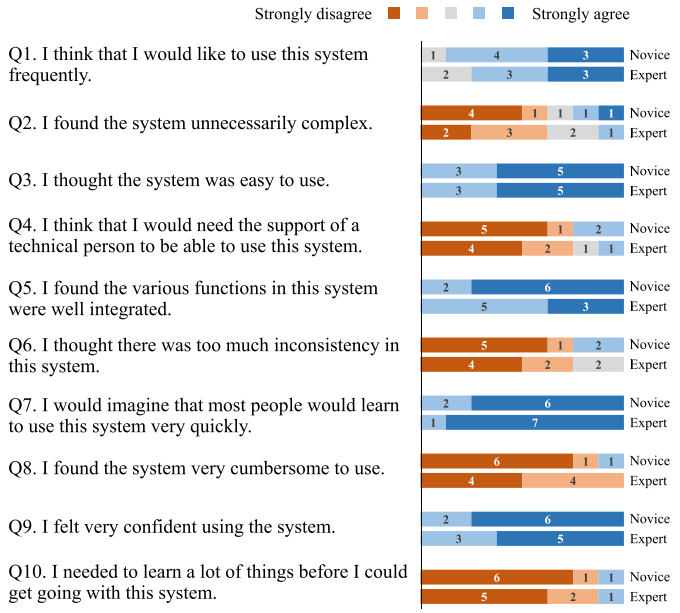


Fig. 13. Ratings of System Usability Scale (SUS) questionnaire [52] that includes both positive and negative statements (odd and even items, respectively). The numbers in the bar represent the number of participants who submitted the same rating.

The runtime comparisons in Table II show that our latent distribution alignment mechanism does not introduce additional runtime overhead compared with the latent blending methods in SmartBrush [12] and Blended LDM [39] without the distribution alignment. The resampling strategies in SDEdit [11] and RePaint [10] lead to longer runtime (about 2×) due to extra sampling steps. The results demonstrate the efficiency of our approach.

D. Evaluation of User Interface

1) Usability and Interaction Speed: We additionally asked the participants in our user study introduced in Section IV-B4 to fill out a System Usability Scale (SUS) questionnaire [52], [53], which includes both positive and negative statements. As shown in Fig. 13, for positive statements (odd items), 100% of users, either novices or experts, agreed that the system was easy to use with well-integrated functions (Q3, Q5). At least 81% felt confident and would like to use it frequently (Q1, Q9). For negative ones (even items), fewer than 19% of users thought the system was complex, cumbersome, and inconsistent (Q2, Q6, Q8). These results indicate the usability of our system.

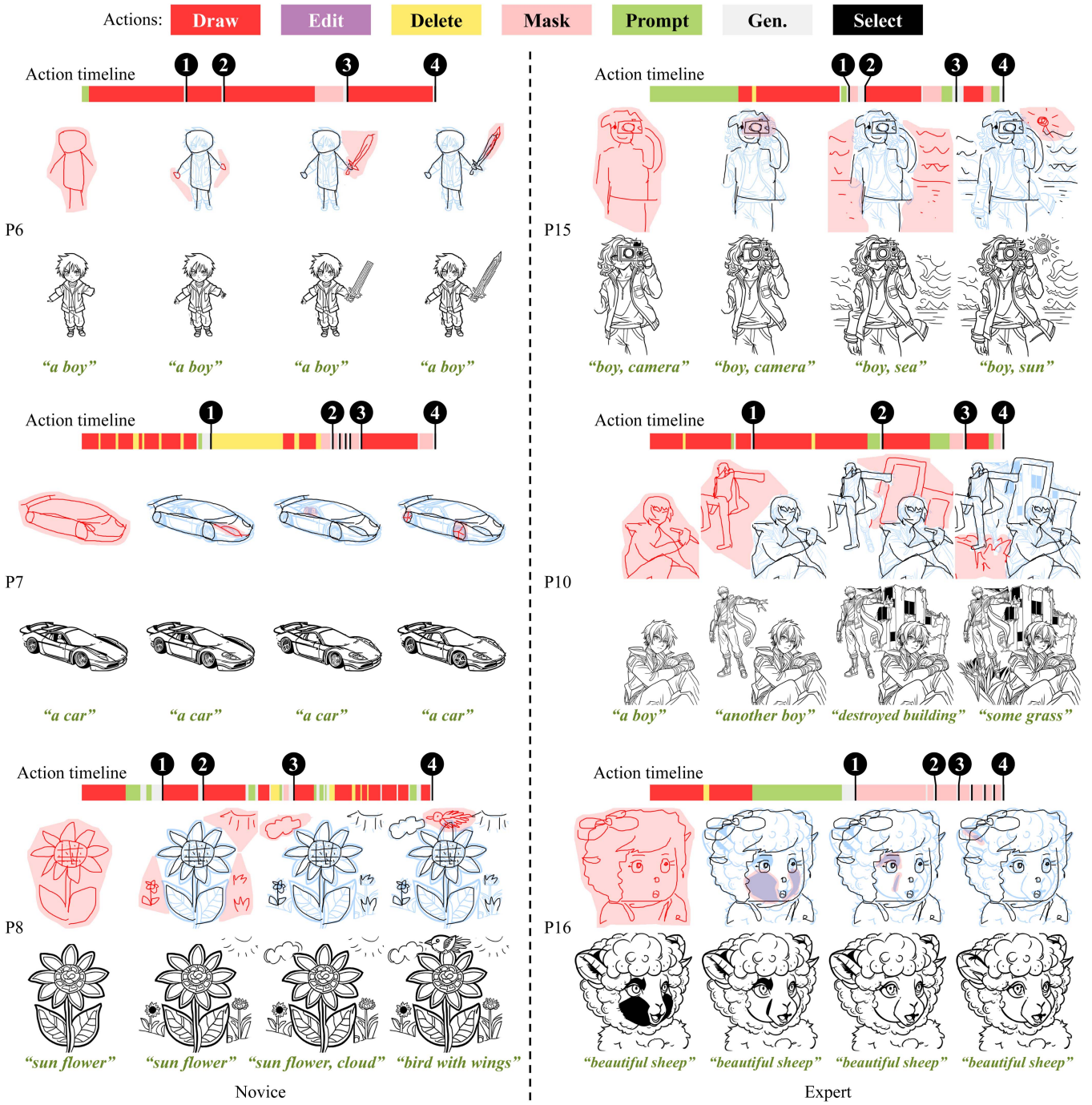


Fig. 14. Support for different creation workflows of participants in the user study. We plot their different actions along a timeline using color coding. The four results of each participant correspond to the orders on the timeline.

Given that each generation practice in our system takes 1.6 seconds on average, 93% of the users were satisfied with the interaction speed, as shown in Q7 of the questionnaire in Fig. 10. In the interview, most users mentioned that the interaction was swift in the experiment.

2) Support for Different Workflows or Requirements: Given our system's stroke, mask, and prompt tools, the users combined them in different ways to create creative usages and workflows in the experiments. As shown in Fig. 14, apart from a common practice that drew strokes first and then generated images with masks and prompts (P7, P8, P10), some users defined prompts first for guiding sketching (P6, P15). Additionally, some users employed masks and prompts without stroke actions for local region refinement. For example, P15 masked the camera for a better generation (the 2nd result). P16 utilized the masks iteratively to remove undesired regions.

While our system is trained exclusively with line art data of single characters, it also supports the creation of animals, objects, multiple characters, and complex scenes, as shown in Figs. 2 and 14. We also validate its robustness to different mask

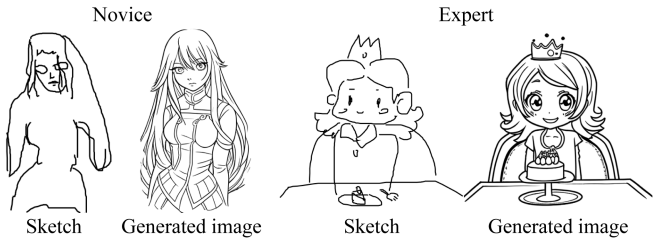


Fig. 15. Left: An acceptable result that does not align well with the sketch. Right: A failure case that misaligns with the input sketch for an existing cartoon character.

sizes or text prompts for fixed sketches. Please refer to the supplementary document for more details. These results indicate that our system is general and practical enough for users with different workflows or requirements to explore possibilities and creativity.

3) Performance Between Novices and Experts: The novice users with little drawing skill tended to draw a single character or a simple and abstract object, as shown in Fig. 14 (P6, P7). Albeit with low alignment with the sketch, such as the left example in Fig. 15, visually reasonable results produced by the system were still acceptable to them. As novice user P3 commented, *"I do not need the appearance alignment for a novice user without drawing skills like me. It met my requirement when the results were reasonable."*

For expert users with strong drawing skills, they liked to draw something more complex, such as a scene, as shown in Fig. 14 (P10, P15). Most of them agreed that the quality of the resulting line art was satisfying according to the questionnaire (Fig. 10-Q2&Q3). In the interview, they also talked about our system's usefulness in two aspects: a) Current results can serve as prototypes in the creation process that can be further refined; b) The system can inspire when users draw something creative.

V. Limitations and Discussion

Users held different attitudes towards the alignment between the generated results and the sketches, as indicated by the mixed ratings of Q6 in Fig. 10. A misaligned but visually reasonable result is acceptable for novice users with little drawing skill. While expert users require high alignment when they draw concrete sketches, especially those for existing cartoon characters, as shown in the right example of Fig. 15. Our system may fail to meet their requirements, because the training dataset does not include such specific characters, and our model attempts to imagine according to the sketch. This inspires us that future systems for sketch-based content creation should be able to control the degree of alignment for different user requirements.

Although our system can generate diverse objects and complex scenes, it is limited to a single drawing style, due to the line art-oriented diffusion model [44] that is fine-tuned with drawings from a particular artist. We think this could be an issue rather than a limitation of our system, because when creating line art in real-world usage, using a consistent style is beneficial to controllability and applicability. Thus, many anime studios typically train with their own data with a specific style. In cases where more styles were required, one could train LoRA models with data of different styles.

The interface's limitation is not allowing pixel-level editing and modification of the generated line art. Another issue is that the quick prompts in the tool panel are fixed now, and a dynamic prompt recommendation according to the generated line art is more useful and friendly to the creation process. These can be future extensions of the system.

VI. Conclusion

We present an interactive and progressive line art generation system called *DoodleAssist*, which is controlled by sketches drawn by users step by step. A latent distribution alignment mechanism is proposed to facilitate the progressive generation built upon a regional latent blending process, improving the regions' transition. An interactive user interface is developed to support line art creation via sketches. We demonstrate our system's effectiveness and generalization ability through comparisons against existing approaches and an in-depth user study of usefulness and usability. The results show that our system helps users, either novices or experts, concretize their intentions and explore possibilities during creation.

References

[1] V. Sarukkai, L. Yuan, M. Tang, M. Agrawala, and K. Fatahalian, "Block and detail: Scaffolding sketch-to-image generation," in *Proc. 37th Annu. ACM Symp. User Interface Softw. Technol.*, 2024, pp. 1–13.
[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10674–10685.
[3] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3813–3824.
[4] E. Barrett, "How to draw everything: Simple sketching and inking step by step lessons," Independently Published, 2021. [Online]. Available: https://www.amazon.com/How-Draw-Everything-Sketching-Beginner/dp/B09MYSRRZW
[5] B. Dodson, *Keys to Drawing*. London, U.K.: Penguin, 1990.
[6] E. Iarussi, A. Bousseau, and T. Tsandilas, "The drawing assistant: Automated drawing guidance and feedback from photographs," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, 2013, pp. 183–192.
[7] D. Dixon, M. Prasad, and T. Hammond, "iCanDraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2010, pp. 897–906.
[8] C. Mou et al., "T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 4296–4304.
[9] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18187–18197.
[10] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using denoising diffusion probabilistic models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11451–11461.
[11] C. Meng et al., "SDEdit: Guided image synthesis and editing with stochastic differential equations," in *Proc. Int. Conf. Learn. Representations*, 2022.
[12] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, "SmartBrush: Text and shape guided object inpainting with diffusion model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22428–22437.
[13] H. Lin, Y. Ye, J. Xia, and W. Zeng, "SketchFlex: Facilitating spatial-semantic coherence in text-to-image generation with region-based sketches," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2025, pp. 1–19.
[14] S.-Y. Chen, W. Su, L. Gao, S. Xia, and H. Fu, "DeepFaceDrawing: Deep generation of face images from sketches," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 1–16, 2020.

[15] T. Portenier, Q. Hu, A. Szabó, S. A. Bigdeli, P. Favaro, and M. Zwicker, "FaceShop: Deep sketch-based face image editing," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, 2018.
[16] X. Peng, J. Koch, and W. E. Mackay, "FusAIn: Composing generative AI visual prompts using pen-based interaction," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2025, pp. 1–20.
[17] Y. Matsui, T. Shiratori, and K. Aizawa, "DrawFromDrawings: 2D drawing assistance via Stroke interpolation with a sketch database," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 7, pp. 1852–1862, Jul. 2017.
[18] Y. J. Lee, C. L. Zitnick, and M. F. Cohen, "ShadowDraw: Real-time user guidance for freehand drawing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–10, 2011.
[19] K. C. Kwan and H. Fu, "Mobi3DSketch: 3D sketching in mobile AR," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–11.
[20] Y. Jiang et al., "Handpainter-3D sketching in VR with hand-based physical proxy," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–13.
[21] Y. Shen, Y. Shen, J. Cheng, C. Jiang, M. Fan, and Z. Wang, "Neural canvas: Supporting scenic design prototyping by integrating 3D sketching and generative AI," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–18.
[22] Q. Su, X. Bai, H. Fu, C.-L. Tai, and J. Wang, "Live sketch: Video-driven dynamic deformation of static drawings," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.
[23] H. J. Smith, Q. Zheng, Y. Li, S. Jain, and J. K. Hodgins, "A method for animating children's drawings of the human figure," *ACM Trans. Graph.*, vol. 42, no. 3, pp. 1–15, 2023.
[24] W. Yang, "Context-aware computer aided inbetweening," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 2, pp. 1049–1062, Feb. 2018.
[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
[26] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using none-quilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
[27] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Representations*, 2021.
[28] A. Voynov, K. Aberman, and D. Cohen-Or, "Sketch-guided text-to-image diffusion models," in *Proc. ACM SIGGRAPH Conf. Proc.*, 2023, pp. 1–11.
[29] B. Peng, J. Wang, Y. Zhang, W. Li, M.-C. Yang, and J. Jia, "ControlNeXt: Powerful and efficient control for image and video generation," 2024, arXiv:2408.06070.
[30] D.-Y. Chen, A. K. Bhunia, S. Koley, A. Sain, P. N. Chowdhury, and Y.-Z. Song, "DemoCaricature: Democratising caricature generation with a rough sketch," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 8629–8639.
[31] S. Koley et al., "It's all about your sketch: Democratising sketch control in diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 7204–7214.
[32] J. Li, N. Gao, T. Shen, W. Zhang, T. Mei, and H. Ren, "SketchMan: Learning to create professional sketches," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 3237–3245.
[33] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[34] Z. Huang, H. Xie, T. Fukusato, and K. Miyata, "AniFaceDrawing: Anime portrait exploration during your sketching," in *Proc. ACM SIGGRAPH Conf. Proc.*, 2023, pp. 1–11.
[35] Y. Nitzan et al., "Lazy diffusion transformer for interactive image editing," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 55–72.
[36] S. Yang, X. Chen, and J. Liao, "Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 3190–3199.
[37] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-or, "Prompt-to-prompt image editing with cross-attention control," in *Proc. Int. Conf. Learn. Representations*, 2023.
[38] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, "DiffEdit: Diffusion-based semantic image editing with mask guidance," in *Proc. Int. Conf. Learn. Representations*, 2023.
[39] O. Avrahami, O. Fried, and D. Lischinski, "Blended latent diffusion," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–11, 2023.
[40] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, "Null-text inversion for editing real images using guided diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6038–6047.
[41] C. Yan, Y. Li, D. Aneja, M. Fisher, E. Simo-Serra, and Y. Gingold, "Deep sketch vectorization via implicit surface extraction," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–13, 2024.
[42] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory*, vol. TIT-29, no. 4, pp. 551–559, Jul. 1983.
[43] K. Kim, "DeepDanbooru," 2024, [Online]. Available: https://github.com/KichangKim/DeepDanbooru
[44] foolkatdesigns, "FoolKat GOD-OF-MONOCHROME," 2024. [Online]. Available: https://civitai.com/models/123631?modelVersionId=142306
[45] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu, "UniPC: A unified predictor-corrector framework for fast sampling of diffusion models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 49842–49869.
[46] H. Mo, E. Simo-Serra, C. Gao, C. Zou, and R. Wang, "General virtual sketching framework for vector line art," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–14, 2021.
[47] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2463–2471.
[48] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[49] Y. Vinker et al., "CLIPasso: Semantically-aware object sketching," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–11, 2022.
[50] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[51] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
[52] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Hum.–Comput. Interaction*, vol. 24, no. 6, pp. 574–594, 2008.
[53] P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, *Usability Evaluation in Industry*. Boca Raton, FL, USA: CRC Press, 1996.