Streamlining Image Editing with Layered Diffusion Brushes

Peyman Gholami Robert Xiao
University of British Columbia
{peymang, brx}@cs.ubc.ca

https://layered-diffusion-brushes.github.io/

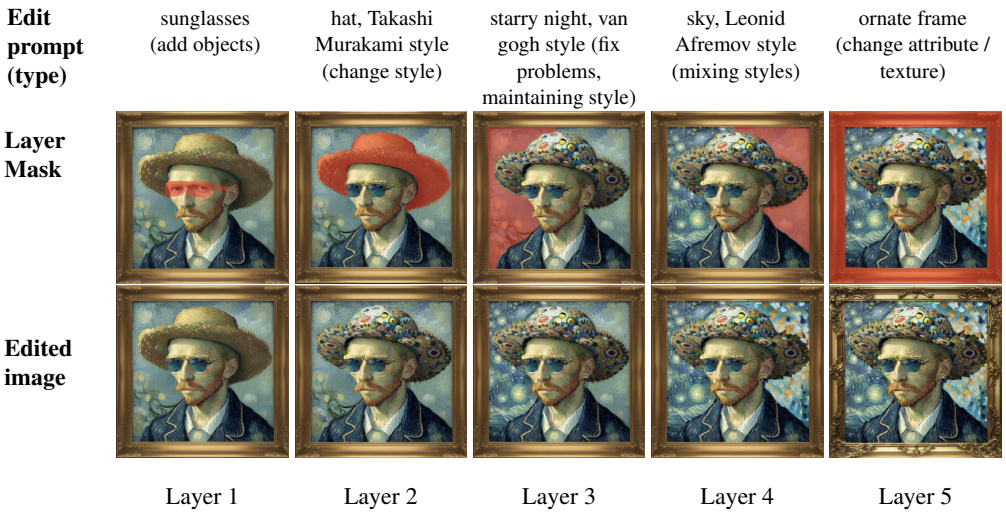

Figure 1. Hierarchical image editing with Layered Diffusion Brushes: LDB is capable of creating and stacking a wide range of independent edits, including object addition, removal, or replacement, colour and style changes/combining, and object attribute modification. Each edit is performed independently, and users are able to switch between the edits seamlessly.

Abstract

Denoising diffusion models have emerged as powerful tools for image manipulation, yet interactive, localized editing workflows remain underdeveloped. We introduce Layered Diffusion Brushes (LDB), a novel training-free framework that enables interactive, layer-based editing using standard diffusion models. LDB defines each "layer" as a self-contained set of parameters guiding the generative process, enabling independent, non-destructive, and fine-grained prompt-guided edits, even in overlapping regions. LDB leverages a unique intermediate latent caching approach to reduce each edit to only a few denoising steps, achieving 140 ms per edit on consumer GPUs. An editor implementing LDB, incorporating familiar layer concepts, was evaluated via user study and quantitative metrics. Results demonstrate LDB's superior speed alongside comparable or improved image quality, background preservation, and edit fidelity relative to state-of-the-art methods across various sequential image manipulation tasks. The findings highlight LDB's ability to significantly enhance creative workflows by providing an intuitive and efficient approach to diffusion-based image editing and its potential for expansion into related subdomains, such as video editing.

1. Introduction

Image editing has undergone transformative advancements with the rise of text-to-image (T2I) generative models, enabling unprecedented creative expression through textual guidance. These models, including Generative Adversarial Networks (GANs) [20], Variational Autoencoders (VAEs), and Denoising Diffusion Models (DMs) [23], have redefined image synthesis and manipulation. Among these, DMs [56] have emerged as the state of the art due to their training stability, high-fidelity outputs, and versatility across tasks like inpainting [37], super-resolution [53], and style transfer [22]. However, despite their capabilities, a critical gap remains: enabling **real-time, localized, and iterative edits** that align with professional workflows, where artists demand precise control over specific regions without

disrupting the global composition.

Existing DM-based editing methods face several core challenges. First, their stochastic nature often necessitates numerous generations to achieve desired results [5]. Second, they lack intuitive mechanisms for layered, non-destructive editing—a cornerstone of tools like Adobe Photoshop [28]—where edits can be independently adjusted, stacked, or removed. Third, while mask-guided approaches enable regional control, they struggle with seamless blending, artifact-free transitions, and real-time feedback. These limitations restrict their adoption in creative pipelines, where rapid iteration and granular control are critical.

To address these challenges, we propose *Layered Diffusion Brushes (LDB)*, a novel framework based on Latent Diffusion Models (LDM) [50] that integrates mask-guided diffusion with a non-destructive layered editing paradigm.

At its core, LDB introduces new noise patterns into the image latents during diffusion process, guided by both the user-specified mask and the edit prompt. This preserves the original context while seamlessly integrating localized edits. We implement an intuitive user interface (UI) with a layering system to support consecutive edits (Fig. 1). Specifically, as key contributions, LDB introduces:

- **Latent Caching for Real-Time Edits:** By reusing intermediate denoising states from initial generation, edits bypass redundant computations and achieve as low as 140 ms per edit on 512×512 images (53× faster than BrushNet [31] using the same consumer GPUs).
- **Non-destructive Layered Editing:** LDB introduces an order-agnostic layering mechanism by defining the concept of layers for DMs, enabling:
  - Region-targeted adjustments with background preservation, using mask-prompt pairs,
  - Stacking, toggling, or deleting layers without cross-interference—even in overlapping regions,
  - Post-hoc revision of edits while preserving underlying content.
- **Seed-Driven Exploration:** Our UI provides familiar "brush" and "scroll" gestures to enable instant exploration of variations by modulating noise seeds, bridging stochastic generation with deterministic refinement and instant feedback.

We validate LDB through extensive experiments and a user study with graphic designers. Quantitatively, LDB outperforms state-of-the-art methods in terms of speed and image quality and is comparable in terms of edit fidelity. The user study revealed superior usability and creativity support in iterative design. Additionally, LDB is a plug-and-play, training-free system adaptable to existing models and applications, and we demonstrate this by applying LDB to the task of video editing.

2. Related Work

2.1. DM-based Image Editing

Image editing is the task of modifying existing images in terms of appearance, structure, or composition, ranging from subtle adjustments to major transformations. Unlike GAN-based approaches [1, 35, 44], which are prone to limitations in inversion stability [49] and localized control [6], diffusion-based methods harness the power of controllable, high-quality DMs in various image-editing tasks, including text and image-driven image manipulation studies [14, 26, 34, 37].

Instruction-based text editing methods [9, 18, 19, 21, 64] typically train DMs on instruction-image pairs. For example, InstructPix2Pix [9] is trained using synthetic pairs from Stable Diffusion [50] and Prompt-to-Prompt [22]. However, expressing nuanced edits solely through text instructions remains challenging, particularly for object-specific style or color changes.

Mask-based methods [4, 5, 14, 63] sample within specified regions. While effective for localized edits, they can introduce unintended global changes, especially problematic in sequential editing, and may struggle with complex edits. For instance, Blended Latent Diffusion's lossy VAE latent space hinders perfect reconstruction even before noise addition [5]. Though a background reconstruction strategy is included, it increases computation and may still yield incoherent results for complex edits. Conversely, our method directly modifies the original latent space, enhancing context preservation and natural blending.

Attention-based editing manipulates cross-attention maps to guide the image generation process toward the desired modifications [22, 45]. These methods generally face challenges in achieving fine-grained edits without unwanted global modifications. Yang et al. [61] attribute unintended changes to inaccurate attention maps and propose attention focusing. Inversion-based methods like ILVR [12], Textual Inversion [16], and DreamBooth [51] focus on context modification while preserving subjects. DDIM inversion converts images to noisy latents, and sampling generates edited results based on prompts. We employ Direct Inversion [30] for efficient real image latent inversion.

Image inpainting involves replacing or restoring the missing regions while maintaining global coherency [60]. Many inpainting works [39, 50, 62, 69] require using a fine-tuned DM specifically designed for inpainting tasks, limiting their applicability. Some, including SmartBrush, which uses object-mask prediction guidance [59], offer more flexibility. PowerPaint [69] introduces learnable task embeddings for improved control. While these models effectively generate new content, they are generally unsuitable for making small, targeted adjustments [4, 37, 52]. Inspired by ControlNet [67], BrushNet [31] builds a decomposed

plug-and-play dual-branch DM, but struggles with real-time interaction due to its computational overhead. In Sec. 4.1 we compare LDB with several inpainting techniques.

2.2. Layered and Sequential Image editing

Layer-based image editing is fundamental in computer graphics [46], and recent works integrate this concept into AI methodologies [6, 54]. Layered representations enable dynamic manipulation of image components, transforming single images into multi-layered structures.

LayeringDiff [33] decomposes images into foreground and background. ParallelEdits [27] uses attention for efficient multi-aspect text edits. MAG-Edit [40] employs a two-layer process with attention injection to a single edit from background. Joseph et al. [29] highlight error accumulation in sequential editing, where artifacts compound across edits. Collage Diffusion [54], built on modified Blended Latent Diffusion [5], employs alpha masks to guide cross-attention and generate harmonized images while respecting scene composition. However, it assumes pre-layered inputs and synthesizes scenes from scratch. In contrast, LDB is training-free, operates directly on existing images, and supports fully independent layers—unlike methods such as [6] that require per-image training.

2.3. Accelerated Generation using Caching

Caching and reusing intermediate features has proven effective for accelerating DM inference through reducing redundant computations. Several works have utilized caching in diffusion transformers (DiTs) for video generation. DeepCache [38] reuses high-level U-Net features in video generation, while AdaCache [32] dynamically adjusts cached residuals based on temporal content. Cache Me If You Can [57] employs block caching by reusing outputs from layer blocks of previous steps during inference. For image generation, Approximate Caching [2] reuses intermediate latents created during prior image generation processes for similar prompts. We employ a similar strategy through caching key latent representations and adapt it specifically for interactive image editing, enabling the real-time feedback that is crucial for creative workflows.

3. Method

We use an LDM-based variant of image generative models and make intermediate adjustments to the latent space, similar to [5, 37]. Therefore, LDB requires no additional training or fine-tuning of the underlying LDM; all modifications are applied during the reverse diffusion process.

We adopt the standard LDM formulation, where image generation begins with a sample from a Gaussian distribution, $Z_0 \sim \mathcal{N}(0, \sigma_{max}^2 I)$ and is iteratively denoised through a sequence of steps N, resulting in a series of latents Z_i corresponding to decreasing noise levels σ_i, where $\sigma_0 = \sigma_{max} > \sigma_1 > \cdots > \sigma_N \approx 0$.

As demonstrated in Fig. 2, the overall LDB pipeline comprises three key stages: **initial image generation (or inversion)**, **latent caching**, and **iterative layered editing.**

For DM-generated images, we first initialize the sample $Z_0 = \epsilon_0$ and noise level σ_0 ($i = 0$). For real images, the initial noise latent is obtained using inversion. We use Direct Inversion [30] for its high speed and comparable performance to other inversion methods, including Null-Text Inversion [43] and Negative-Prompt Inversion [42]. The noisy sample then undergoes the diffusion process, caching certain intermediate latents to facilitate editing.

3.1. Latent Caching

To enable rapid, interactive editing with instant exploration and feedback, we employ latent caching to reuse intermediate representations in subsequent steps, minimizing redundant computations. We store two key intermediate latents:

- **Regeneration Latent $\mathbf{Z_r}$:** At diffusion step $r = N - n$, where N is the total number of diffusion steps for initial image generation and n is the number of editing steps, we cache the latent Z_r, which serves as the starting point for all subsequent edits. By reusing Z_r, we avoid recomputing the initial denoising steps for each new edit, significantly speeding up the editing process (from N denoising steps to n). Effectively, Z_r represents a partially denoised latent state that retains the global image structure but is still malleable enough to accommodate localized edits.
- **Blending Latent $\mathbf{Z_b}$:** We cache the latent at diffusion step b which is specifically used for the layer merging process (Algorithm 1, line 7). We set $b = N - 2$ for maximum background preservation (as discussed in Sec. 4.3). Z_b represents a more denoised latent compared to Z_r, capturing more refined image details while still allowing for seamless blending of new edits into the existing image context. Utilizing this cached blending latent ensures smoother integration of edits and reduces visual artifacts at layer boundaries during the merging process.

3.2. Layered Diffusion Brushes Editing

To initiate an edit, the algorithm begins by generating a new noise pattern $Z_0' = \epsilon_k'$, sampled from $\mathcal{N}(0, \sigma^2 I)$ using a different seed S', and scaling it to match the variance of the cached latent Z_r. This ensures that the additive noise stays in a reasonable range from the latent for editing, preventing visual artifacts. Z_0' is then added to the regeneration latent Z_r, controlled by the mask m and strength α.

In the editing stage, at step b, a new noisy sample is merged with the cached blending latent using the strength control and the mask, resulting in Z_b'. Subsequently, the new latent is progressively denoised from steps b through N and processed through the VAE to output edited image I'.

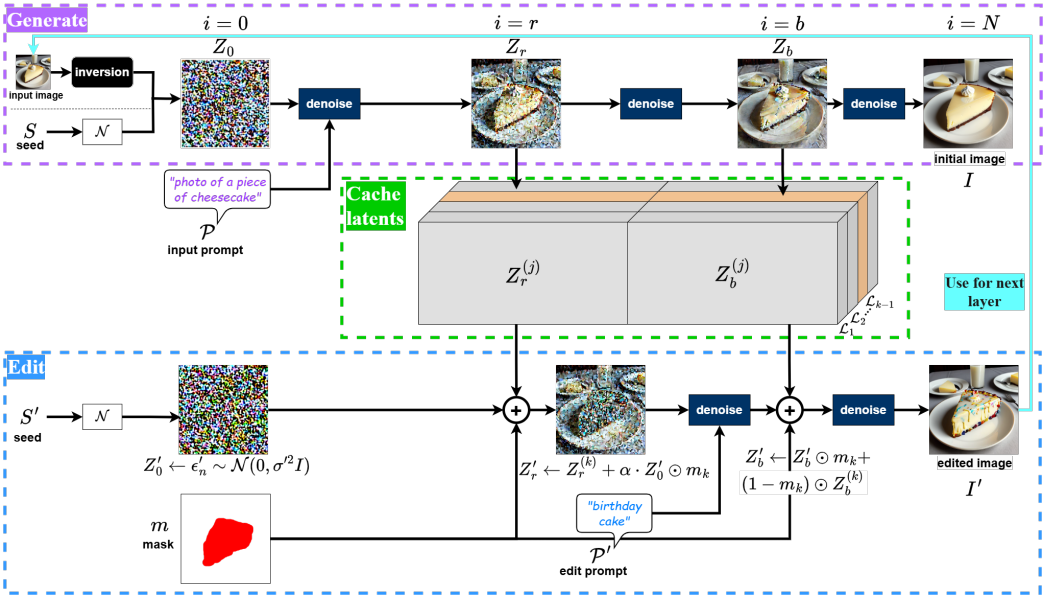

Figure 2. Overview of the Proposed Method: The top box shows standard DM-based image generation from noisy latent Z_0 and prompt $\mathcal{P}$. The middle section depicts the latent caching module, storing and retrieving intermediate latents for different layers. The bottom box illustrates the editing process: a new noise sample S' merges with the original latent at step r using mask m and strength control α. Diffusion continues until step b, where modified and cached latents blend to generate the final edited image.

Algorithm 1 presents the pseudocode for the editing process for a single layer (for simplicity):

Algorithm 1: LDB editing process (single layer)

Input : Edit prompt $\mathcal{P}'$, Mask $m \in [0,1]^{H \times W}$, Random seed S', Strength α, Number of edit steps n, Regeneration latent Z_r, Blending latent Z_b

Output: Edited latent Z'_N

$Z'_0 \leftarrow \epsilon'_{n_k} \sim \mathcal{N}(0, \sigma^2 I)$ // sampled using seed S'

$Z'_0 \leftarrow \sqrt{Var(Z_r)} \cdot Z'_0$ // scale new sample

$Z'_0 \leftarrow Z_r + \alpha \cdot (Z'_0 \odot m)$ // noise injection

for $i = 0, 1, \ldots, n$ **do**

  $Z'_{i+1} \leftarrow DM(Z'_i, \mathcal{P}', i, S')$

  if $i == b$ **then**

    $Z'_b \leftarrow Z'_b \odot m + Z_b \odot (1 - m)$ // blending

end

Return Z'_N

3.3. Layer Formulation

Unlike prior works that rely on transparent decomposable layers [66] or explicit object segmentation [54], we redefine a layer as a self-contained set of reproducible parameters that govern localized edits. For layer $\mathcal{L}_k$, we formalize this as a generalized version of parameters in Algorithm 1:

$$\mathcal{L}^{(k)} = \left(\mathbf{S}'^{(k)}, \mathbf{m}^{(k)}, \mathbf{v}^{(k)}, \mathbf{Z}_r^{(k)}, \mathbf{Z}_b^{(k)}, \alpha^{(k)}, n^{(k)}, \mathcal{P}'^{(k)}, j\right) \quad (1)$$

- $\mathbf{S}'^{(k)} \in \mathbb{Z}^+$: Seed space for stochastic variations
- $\mathbf{m}^{(k)} \in [0,1]^{H \times W}$: Edit mask
- $\mathbf{v}^{(k)} \in \{0,1\}$: Visibility state
- $\mathbf{Z}_r^{(k)}, \mathbf{Z}_b^{(k)} \in \mathbb{R}^{C \times H \times W}$: Regeneration/blending latents
- $\alpha^{(k)} \in [0,1]$: Layer strength value
- $\mathbf{n}^{(k)} \in [0, N]$ Number of denoising steps
- $\mathcal{P}'^{(k)}$: Edit prompt
- $j \in \mathbb{Z}^+$: Index of last layer index.

Notably, within a given layer $\mathcal{L}_k$ with previous layer $\mathcal{L}_j$, the cached latents $\mathbf{Z}_r^{(j)}$ and $\mathbf{Z}_b^{(j)}$ inherently incorporate the cumulative edits from all preceding layers. This is because edits to layer $\mathcal{L}_k$, are applied to the already edited output of layer $\mathcal{L}_j$ which serves as the input to the diffusion process and the algorithm always keeps the last layer updated. Therefore, any modification in a previous layer automatically propagates through the subsequent layers. By defining Φ as a single-layer latent generation and caching step as:

$$(Z_r^{(k)}, Z_b^{(k)}) = \Phi(\mathcal{L}^{(k)}, \mathcal{L}^{(j)}) \quad (2)$$

in essence, if a given layer $\mathcal{L}^{(i)}$ (where $i < k$) is removed or its visibility $\mathbf{v}^{(i)}$ is toggled, the operator Φ will be recursively invoked to recreate all latents for layers from $\mathcal{L}^{(i)}$ to $\mathcal{L}^{(k)}$. This recomputation, accelerated by latent caching, is automatically triggered and typically completes within milliseconds to a few seconds, depending on the number of layers. This design allows edits to remain independent yet seamlessly integrated into the final composition.

3.3.1. Overlapping Regions

A key advantage of layered editing in LDB is the ability to create overlapping edits, where one layer can partially or fully modify areas affected by earlier layers. This requires careful handling of each layer's regeneration latent, Z_r, to ensure that changes in visibility or content from higher layers are accurately reflected in subsequent layers, even in overlapping regions.

By default, all layers use the initial image's latent (Z_r) as their regeneration latent. However, this approach fails to account for overlapping edits from preceding layers. To address this, when processing a layer k, we compute its regeneration latent by inverting the output image of the previous layer ($I'^{(k)}$) as shown using the feedback arrow on Fig. 2. This inversion yields $Z_0^{(k)}$, which is then sent through the generation stage in LDB. Both $Z_r^{(k)}$ and $Z_b^{(k)}$ are cached for efficient processing (as shown in Fig. 3).

This mechanism enables precise control and seamless integration of edits across overlapping regions. Changes to any layer propagate correctly without introducing artifacts, offering flexibility and fine-grained control.

3.4. User-Interface and Interaction Design

To develop a practical tool for artists and designers, we designed an custom UI that balances control and simplicity. The UI allows users to generate, upload, and edit images, manage layers, and adjust parameters seamlessly. Two interaction modes streamline edits (Fig. 4):

Box Mode: Users can click or drag on the image to move a resizable square mask around. This option enables a quick and interactive exploration of how various parts of the image will change in response to a given set of editing settings (prompt and strength), simply by moving the cursor.

Custom Mask Mode: Users can draw free-form masks over the desired around and navigate between new generation samples by scrolling the mouse up or down while hovering over the image, allowing them to rapidly explore variations on their edit.

We propose a workflow where users first position edits spatially using Box Mode, then refine mask geometry and appearance details via Custom Mask Mode.

Layering capabilities include stacking, visibility toggling, and deletion. Each layer is independently modifiable. Detailed information on the UI design user interactions and a demo video can be found in supplementary material.

4. Experiments

4.1. User Study

We conducted a user study in order to evaluate the effectiveness of LDB for providing targeted image fine-tuning, using two other well-known existing image editing tools, InstructPix2Pix (IP2P) [9] and Stable Diffusion Inpainting (SDI) [50] as baselines for comparison.

We recruited a cohort of seven expert participants with extensive experience in using image editing software. As part of our selection criteria, we ensured that all had at least a basic level of familiarity with AI image generation techniques [3, 41] and were regular users of editing software, such as Adobe Photoshop [28] for creating visual art.

4.1.1. Study Procedure and Task Description

Each user engaged in two types of tasks: free-form tasks where users generated an image for editing using a fixed prompt and seed (type 1), and pre-determined tasks where the user worked with existing real images from the MagicBrush dataset [65] (type 2).

For the type 1 tasks, we selected specific types of edits that showcase various functionalities and capabilities of the system, including:

1. Stack layers and create sequential edits (draw with LDB)
2. Modify attributes and features of objects
3. Correct image imperfections and errors
4. Enhance discernibility of similar objects
5. Target specific regions for style transfer, refine aesthetics

Type 2 tasks were more structured, with the mask, edit prompt, and input images provided by the dataset. The dataset provides manually annotated masks and instructions for each edit. We selected a subset of 35 input images, each containing up to three layers of edits. Users refined masks/parameters if necessary and completed editing tasks.

Figure 5, second row, shows example edits generated by the participants. Additional examples are provided in the supplementary material. As shown, LDB produces targeted edits that integrate seamlessly with the images.

4.1.2. Evaluation Survey Results

The participants completed a three stage evaluation survey following the image editing tasks. The first part included a System Usability Scale (**SUS**) form to rate the usability, ease of use, design, and performance of each method. SUS is a standard usability evaluation survey widely used in user-experience literature [8]. Overall, participants indicated that they are more likely to use LDB compared to IP2P and SDI, and that they find it the easiest tool to use. **LDB** obtained a SUS score of **80.35%**, while **IP2P** and **SDI** achieved a SUS of **38.21%** and **37.5%** respectively.

The SUS survey was followed by a Creativity Support Index [11] survey to evaluate the system's degree of creative work support. Participants expressed positivity towards LDB, indicating that it enhanced their enjoyment, exploration, expressiveness, and immersion, while also deeming the results worth their effort. Lastly, the survey was followed by a semi-structured interview where participants appreciated the intuitiveness, ease of use and versatility of

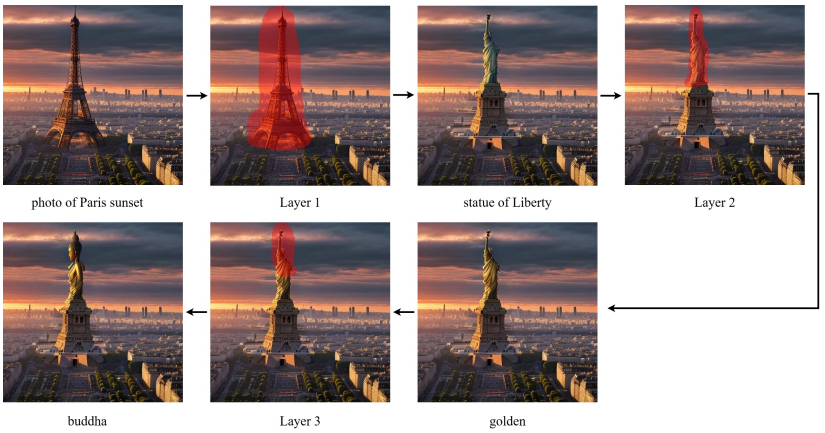

Figure 3. Overlapping edit regions in LDB: overlapping edits enable complex, interacting modifications. For example, one layer can adjust color while another changes shape, with the final result combining both.

(a) Box option with moving cursor (b) Custom mask option with mouse scroll

Figure 4. Box and Custom Mask Options: In box mode, users click the target region's center to generate edits within the specified area and can drag the box to explore variations instantly. In custom mask mode, users draw a mask over the desired region and adjust the seed using the mouse wheel or scrolling gestures to generate new variations.

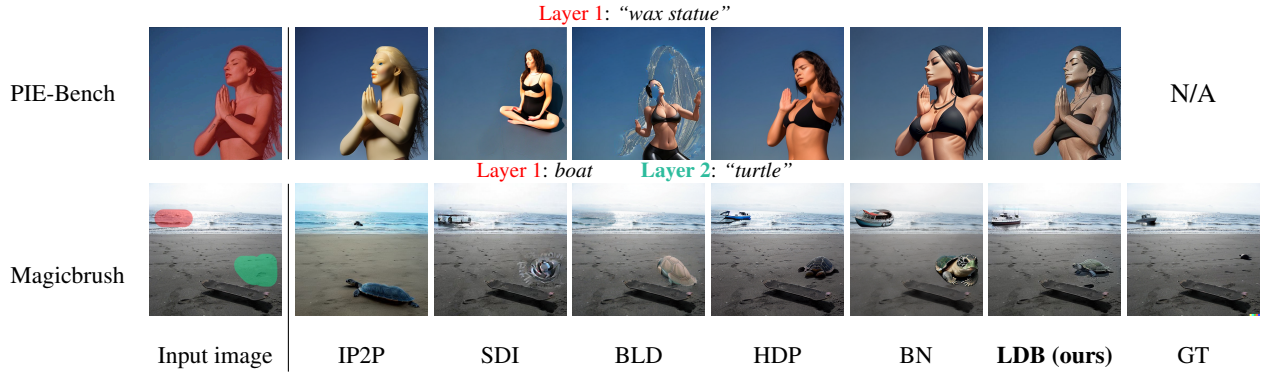

Figure 5. Qualitative editing results on PIE-Bench (top) and MagicBrush (bottom) benchmarks using different methods. Edit prompts are presented on top of each row. More examples available in supplementary material.

LDB. Further details about the study, interview, results, and discussion can be found in supplemental material.

4.2. Quantitative Analysis

To quantitatively evaluate the performance of LDB, we employed a comprehensive suite of metrics, aligning with established practices in image editing evaluation.

Specifically, for text-image alignment, we used CLIP Score (CS) [47] for global alignment, CS-L for masked-region alignment, and CS-D [17] for consistency between image and caption changes in CLIP space.

We adopted Learned Perceptual Image Patch Similarity (**LPIPS**) [68] and Peak Signal-to-Noise Ratio (**PSNR**) [24] for evaluating content preservation and pixel-level fidelity in unmasked regions. Furthermore, to gauge overall image quality and aesthetic appeal, we incorporated Aesthetic Score (**AS**) [55], Image Reward (**IR**), and Human Preference Score V2 (**HPS**) [58], the latter two reflecting human-aligned preferences.

We compared **LDB** against a diverse set of state-of-the-art editing and inpainting methods, including InstructPix2Pix (**IP2P**) [9], Stable Diffusion Inpainting (**SDI**) [50], HD-Painter (**HDP**) [39], BrushNet (**BN**) [31], and Blended Latent Diffusion (**BLD**) [5], on two benchmarks: MagicBrush [65] and PIE-Bench [30]. For MagicBrush, we also report results on the provided ground truth (**GT**) images.

Quantitative results are summarized in Tab. 1. All methods were evaluated using their default editing settings, except for LDB, IP2P, and SDI on the MagicBrush benchmark, where we used user-edited images from our user study for consecutive edits. Inference times denote average per-edit durations, measured on a single NVIDIA RTX 4090 GPU with $N = 25$ diffusion steps for baseline methods and $n = 8$ for LDB.

4.3. Ablation Study

We perform three ablation studies for two main components of the LDB caching mechanism, *i.e.* the caching timesteps for the regeneration latent (r), and the blending latent (b). We also ablate and discuss the effect of strength control α

Benchmark	Method	Image Quality			Masked Region Preservation		Text Alignment			Time (s)
		IR $_{\times 10}$ ↑	HPS $_{\times 10^2}$ ↑	AS ↑	PSNR ↑	LPIPS $_{\times 10^2}$ ↓	CS ↑	CS-L ↑	CS-D $_{\times 10^2}$ ↑	(per edit) ↓
MagicBrush	IP2P	-62.83	21.16	5.29	7.28	15.07	29.39	22.01	6.64	1.72
	SDI	-39.21	20.88	5.48	12.20	8.70	30.08	22.15	4.11	1.84
	HDP	-20.69	**23.27**	5.44	12.05	**6.13**	31.01	22.06	9.89	12.85
	BN	**-0.04**	22.57	**5.73**	11.55	8.75	**31.16**	**22.17**	**12.92**	7.49
	BLD	-24.10	**22.80**	5.48	**12.64**	**6.94**	30.64	21.99	**10.05**	**1.41**
	GT	-1.93	22.62	5.36	17.64	2.30	30.75	22.14	9.78	NA
	Ours	**7.74**	22.65	**5.74**	**12.85**	7.05	**31.04**	**22.07**	9.54	**0.26**
PIE-Bench	IP2P	-40.73	23.12	5.76	172.18	15.47	30.00	**22.79**	14.27	1.83
	SDI	43.46	25.77	6.00	**181.58**	**3.89**	31.24	22.71	14.83	3.36
	HDP	39.02	25.92	6.01	178.84	4.62	31.08	22.73	16.20	13.44
	BN	72.77	**26.66**	**6.17**	177.07	8.67	31.50	**22.80**	**16.88**	7.51
	BLD	**50.68**	26.36	6.11	180.85	4.19	**31.35**	22.78	**17.22**	**1.47**
	Ours	**86.02**	**26.60**	**6.51**	**184.57**	**1.91**	**31.66**	22.76	16.74	**0.25**

Table 1. Quantitative results on MagicBrush and PIE-Bench. Metrics are grouped into Image Quality, Masked Region Preservation, and Text Alignment. ↑ indicates higher is better; ↓ indicates lower is better. The **best** and **second-best** scores are highlighted.

and its relationship with n in supplementary material.

4.3.1. Ablation on Regeneration Latent Step

The timestep r for caching the regeneration latent is critical, as it dictates the extent of possible modifications during the regeneration process. We performed an ablation study by varying r while holding the total diffusion steps N constant. This variation in r implicitly changes the number of regeneration steps (n) and necessitates adjustments to the strength parameter accordingly. Qualitatively, as shown in Fig. 6, excessively small r values lead to incoherent edits and noticeable artifacts due to insufficient blending with the original image. Conversely, large r values limit the model's ability to modify the masked region, resulting in minimal changes and preserving the original content.

Quantitatively, we observe that smaller r steps (*e.g.* $r = 2$) yield higher LPIPS (0.04) and low PSNR (27.03), indicating poor image quality and fidelity. Edit fidelity scores such as CS-L also confirm that larger r steps result in lower scores (22.98), suggesting ineffective edits within the masked region. The HPS index demonstrates a higher score for mid-range steps (0.33, $r = 12$) compared to both ends of the spectrum (0.29, $r = 23$), highlighting a performance sweet spot for intermediate r values. Detailed metric graphs are available in the supplemental material.

4.3.2. Ablation on Blending Latent Step

The blending latent step, controlled by the parameter b, determines when the cached regeneration latent is blended back into the diffusion process and is crucial for seamless integration of the edited region with the original image and preserving background. We conduct an ablation study by varying b while keeping r and N fixed. Fig. 7 qualitatively demonstrates the effect of different b values.

When b is small, the blending process starts prematurely, causing the edit to bleed into the background and distorting the original image context. Conversely, larger b values, representing late blending, effectively preserve the background integrity while still allowing for meaningful edits within the masked region.

Quantitatively, smaller b values ($b = n$) lead to higher LPIPS (0.17) and lower PSNR (11.61), indicating worse background preservation. Edit fidelity scores (CS-L) within the masked remained stable across the spectrum while CS-D improves at larger b (0.32 at $b = N-1$), reflecting better edit alignment. These findings indicate that later blending is preferable, leading us to select $b = N - 2$ in the LDB algorithm to prioritize background preservation while maintaining effective localized editing. Further details and metric plots are available in the supplementary material.

5. Discussion

Our experiments demonstrate that LDB establishes new benchmarks for speed and workflow adaptability in diffusion-based image editing. Key findings include:

Enhanced Control via Layering: LDB's layered design enables creating non-destructive refinements as well as iterative complex compositions. Participants highlighted how this mirrors professional editing tools like Photoshop [28].

Speed and Efficiency: LDB achieves remarkable speed, 53× faster than BrushNet (evaluated on the same hardware), crucial for interactive editing. We observe that reducing diffusion steps to as few as $n = 4$ maintains reasonable quality (HPS: 0.34, CS-D: 0.35), yielding a latency of **140ms** per edit. User studies confirm *instant feedback* as a key advantage, enabling rapid iteration (tens of variations per minute *vs*. 1-2 for baselines). This speed results from

Figure 6. Ablation study on regeneration latent step r (increasing left to right). Small r results in strong prompt adherence ("cat") but introduces artifacts. Large r (near N) leads to insufficient modification, retaining the original "dog". An intermediate r achieves the best balance of edit fidelity and background preservation.

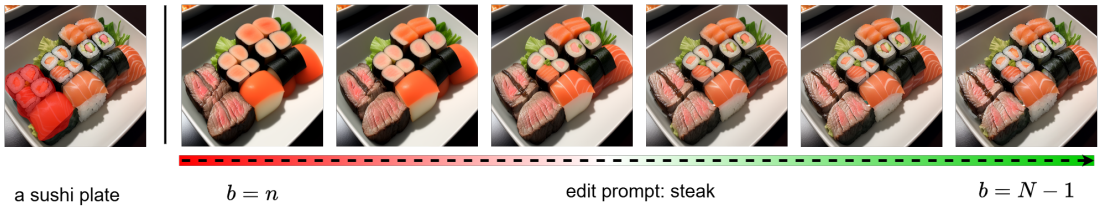

Figure 7. Ablation study on blending latent step b (increasing left to right). The prompt "steak" is applied to an image of "sushi plate" while increasing b from left to right. At $b = n$ (left), the edit disrupts the original structure, affecting unmasked regions. As b approaches N (right), background preservation improves, and edits blend more seamlessly.

efficient latent caching (Sec. 3.1), minimizing computation and memory overhead (∼1.25 MB for 10 layers).

Quantitative Performance: LDB demonstrates a superior combination of speed, image quality, and edit fidelity across both benchmarks. On the PIE-Bench dataset, LDB achieves the best performance in six key metrics, excelling in human preference (HPS = 86.02), background preservation (LPIPS = 1.91), and text alignment (CS = 31.66), while also being the fastest method by a significant margin. This highlights its ability to generalize across a diverse set of editing tasks while maintaining high speed. Similarly, on the MagicBrush benchmark, LDB delivers strong performance with highest score in crucial metrics such as IR, AS, and PSNR. While BrushNet shows a slight advantage in some text alignment metrics, its practical usability is hindered by substantially slower runtime.

5.1. Limitations and Future Work

Brush strength (α) and diffusion step count (n) coupling (Fig. 12) still requires minor user tuning across scenarios. Although preset profiles partially address this, future work could explore adaptive parameter tuning mechanisms to further improve usability. Moreover, semantically implausible edits (*e.g.* placing a boat in the sky) remain challenging due to inherent biases within diffusion models. Integrating techniques like semantic guidance could expand plausible edit ranges. Finally, responsible deployment necessitates robust watermarking [15] and provenance tracking to mitigate misuse and ensure transparency.

5.2. Broader Applications

LDB's training-free design only requires a standard iterative denoising process, which allows seamless integration into diverse diffusion models and applications requiring rapid editing. We validated this by integrating LDB to other commonly used methods, including DiT-based text-to-image (*e.g.*, PixArt-α [10]) and video generation models [7] without any model-specific tuning.

Traditional diffusion-based video editing typically propagates edits from the first frame using additional supervision (*e.g.* optical flow [36]), risking temporal inconsistencies. LDB's high fidelity background preservation and efficiency naturally address these issues.

We demonstrate preliminary success integrating LDB with Stable Video Diffusion (SVD) [7], editing the first frame and applying LDB's latent caching across frames for fast consecutive edits (see supplementary material, Fig. 15). This approach opens avenues for accelerated video manipulation, 3D asset editing, and collaborative design platforms.

5.3. Conclusion

LDB reimagines diffusion-based editing through latent caching and non-destructive layering, achieving unmatched speed and control. Quantitative results and user study show superior performance in image preference, edit fidelity, time, and usability. By bridging interactive editing with high-fidelity generative models, LDB can empower artists to iterate fluidly while maintaining artistic intent.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would also like to thank Professors Leonid Sigal and Kwang Moo Yi for their guidance and support throughout this project.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021. 2

[2] Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving {Text-to-Image} diffusion models. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 1173–1189, 2024. 3

[3] Ideogram AI. Ideogram: Text-to-image generation platform. https://ideogram.ai, 2025. 5

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2

[5] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 2, 3, 6

[6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 707–723. Springer, 2022. 2, 3

[7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 8, 2

[8] John Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 1996. 5, 6

[9] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 2, 5, 6

[10] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 8

[11] Erin Cherry and Celine Latulipe. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(4):1–25, 2014. 5, 9

[12] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 2

[13] Civitai. Dreamshaper - 7 — stable diffusion checkpoint, 2024. Accessed on Feb 19, 2024. 4

[14] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[15] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 8

[16] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[17] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 6

[18] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023. 2

[19] 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*, pages 12709–12720, 2024. 2

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[21] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. *arXiv preprint arXiv:2312.10113*, 2023. 2

[22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1, 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[24] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 6

[25] 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. 1

[26] Minghui Hu, Yujie Wang, Tat-Jen Cham, Jianfei Yang, and Ponnuthurai N Suganthan. Global context with discrete diffusion in vector quantised modelling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11502–11511, 2022. 2

[27] Mingzhen Huang, Jialing Cai, Shan Jia, Vishnu Suresh Lokhande, and Siwei Lyu. Paralleledits: Efficient multi-aspect text-driven image editing with attention grouping. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[28] Adobe Inc. Adobe photoshop (2024 version). https://www.adobe.com/products/photoshop.html, 2024. 2, 5, 7, 9

[29] K. J. Joseph, Prateksha Udhayanan, Tripti Shukla, Aishwarya Agarwal, Srikrishna Karanam, Koustava Goswami, and Balaji Vasan Srinivasan. Iterative multi-granular image editing using diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8107–8116, 2024. 3

[30] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023. 2, 3, 6

[31] 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. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 2, 6

[32] Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. *arXiv preprint arXiv:2411.02397*, 2024. 3

[33] Kyoungkook Kang, Gyujin Sim, Geonung Kim, Donguk Kim, Seungho Nam, and Sunghyun Cho. Layeringdiff: Layered image synthesis via generation, then disassembly with generative knowledge. *arXiv preprint arXiv:2501.01197*, 2025. 3

[34] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 2

[35] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 693–702, 2021. 2

[36] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2024. 8

[37] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 1, 2, 3

[38] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024. 3

[39] Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: High-resolution and prompt-faithful text-guided image inpainting with diffusion models. *arXiv preprint arXiv:2312.14091*, 2023. 2, 6

[40] Qi Mao, Lan Chen, Yuchao Gu, Zhen Fang, and Mike Zheng Shou. Mag-edit: Localized image editing in complex scenarios via mask-based attention-adjusted guidance. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6842–6850, 2024. 3

[41] Inc. Midjourney. Midjourney: Ai image generation. https://www.midjourney.com, 2025. 5

[42] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 3

[43] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 3

[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*, pages 1–11, 2023. 2

[45] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, New York, NY, USA, 2023. Association for Computing Machinery. 2

[46] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984. 3

[47] 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*, pages 8748–8763. PmLR, 2021. 6

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 6

[49] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2

[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*, pages 10684–10695, 2022. 2, 5, 6

[51] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine

tuning text-to-image diffusion models for subject-driven generation, 2023. 2

[52] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[53] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[54] Vishnu Sarukkai, Linden Li, Arden Ma, Christopher Ré, and Kayvon Fatahalian. Collage diffusion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4208–4217, 2024. 3, 4

[55] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 6

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[57] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6211–6220, 2024. 3

[58] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 6

[59] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22428–22437, 2023. 2

[60] Zishan Xu, Xiaofeng Zhang, Wei Chen, Minda Yao, Jueting Liu, Tingting Xu, and Zehua Wang. A review of image inpainting methods based on deep learning. *Applied Sciences*, 13(20):11189, 2023. 2

[61] Fei Yang, Shiqi Yang, Muhammad Atif Butt, Joost van de Weijer, et al. Dynamic prompt learning: Addressing cross-attention leakage for text-based image editing. *Advances in Neural Information Processing Systems*, 36:26291–26303, 2023. 2

[62] 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*, pages 3190–3199, 2023. 2

[63] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023. 2

[64] Ziyun Zeng, Hang Hua, Jianlong Fu, Jiebo Luo, et al. Promptfix: You prompt and we fix the photo. *Advances in Neural Information Processing Systems*, 37:40000–40031, 2025. 2

[65] 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*, 36, 2024. 5, 6

[66] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 4

[67] 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*, pages 3836–3847, 2023. 2

[68] 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*, pages 586–595, 2018. 6

[69] 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. In *European Conference on Computer Vision*, pages 195–211. Springer, 2024. 2