# SmartEraser: Remove Anything from Images using Masked-Region Guidance

Longtao Jiang[1,*] Zhendong Wang[1,*] Jianmin Bao[2,*†♡]
Wengang Zhou[1,†] Dongdong Chen[2] Lei Shi[2] Dong Chen[2] Houqiang Li[1]
[1]University of Science and Technology of China [2]Microsoft Research Asia

https://longtaojiang.github.io/smarteraser.github.io/

Figure 1. Object removal results generated by SmartEraser using user-provided masks. SmartEraser not only generates photorealistic images with the target objects removed but also better preserves the surrounding context around the removed objects.

## Abstract

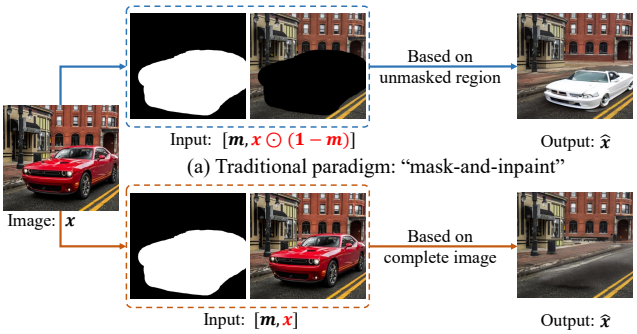
*Object removal has so far been dominated by the "mask-and-inpaint" paradigm, where the masked region is excluded from the input, leaving models relying on unmasked areas to inpaint the missing region. However, this approach lacks contextual information for the masked area, often resulting in unstable performance. In this work, we introduce SmartEraser, built with a new "**removing**" paradigm called Masked-Region Guidance. This paradigm retains the masked region in the input, using it as guidance for the removal process. It offers several distinct advantages: (a) it guides the model to accurately identify the object to be removed, preventing its regeneration in the output; (b) since the user mask often extends beyond the object itself, it aids in preserving the surrounding context in the final result. Leveraging this new paradigm, we present Syn4Removal, a large-scale object removal dataset, where instance segmentation data is used to copy and paste objects onto images as removal targets, with the original images serving as ground truths. Experimental results demonstrate that SmartEraser significantly outperforms existing methods, achieving superior performance in object removal, especially in complex scenes with intricate compositions.*

## 1. Introduction

With the rapid advancement of generative models [10, 11, 15, 32, 34, 35, 37, 40], image editing has garnered significant attention for its broad applications. Among various editing tasks, object removal [9, 55] is especially prominent, empowering users to seamlessly eliminate unwanted elements while maintaining the realism of original images. This functionality, essential for removing distracting objects in photos, has become a key feature in widely used applications like Photoshop, Google Photos, Microsoft Designer.

Currently, most object removal methods [1, 9, 30, 39, 55] adopt a "mask-and-inpaint" paradigm, where the masked region is excluded from the input and typically filled with a neutral placeholder (*e.g.*, black). The model then inpaints the masked region based on the surrounding content. However, we identify two primary issues with this approach. *First*, this strategy often generates unintended objects within the masked region. Lacking precise discrimination between removal targets and other content, these methods rely heavily on background context, sometimes leading to the unintentional inpainting of new objects—such as adding a new car on the road rather than removing the original one, as illustrated in Figure 2(a). *Second*, user-

* Equal contribution. † Corresponding authors. ♡ Project leader.




(a) Traditional paradigm: "mask-and-inpaint"

(b) Our paradigm: Masked-Region Guidance

Figure 2. Comparison of the "*mask-and-inpaint*" paradigm and the proposed *Masked-Region Guidance* paradigm for object removal. Unlike the *mask-and-inpaint* approach, our paradigm uses the masked regions as critical guidance for object removal.

defined masks frequently exceed the target object, requiring the "mask-and-inpaint" approach to synthesize these extended region. This will inadvertently modify nearby context, reducing visual coherence.

To address these limitations, we introduce a novel paradigm for object removal called *Masked-Region Guidance*. The core idea is that the masked region should not be excluded but rather utilized as critical guidance during the removal process. Our method is straightforward: instead of replacing the masked region with a placeholder, we retain the original image as input, with the masked region indicated by a mask input, as in existing methods. As illustrated in Figure 2(b), this paradigm enables the model to accurately identify the target object, preventing unintended regeneration in the output and effectively preserving the surrounding context of the target object in the final result.

However, our new paradigm cannot be directly implemented with existing object removal data construction methods. Typically, their training data are created by masking parts of the image, and the model is trained to predict the masked content. If applying this approach to our proposed *Masked-Region Guidance* paradigm, the model could exploit a shortcut by simply replicating the masked content from the input, due to the masked region being included in the input. Thus, a dataset consisting of triplets of an input, a mask, and the removal result is essential for our paradigm. Unfortunately, existing related datasets [23, 36, 45, 48] either contain a limited number of unique scenes or rely on inpainting models to generate pseudo-removed results.

To solve this, we introduce a synthetic technique to create training data specifically for object removal. Our approach involves pasting object instances from various images onto different background images to form the input images, with the pasted instance masks serving as the input masks and the original background images designated as ground truths. Using this method, we generate Syn4Removal, a large-scale dataset comprising triplets of real background images, masks, and backgrounds with pasted objects. The design of Syn4Removal provides diverse scenes and supports effective training under our new paradigm, encouraging the model to accurately learn object removal without shortcuts.

To make Syn4Removal suitable for training object removal models, we design a pipeline to generate high-quality data. First, we filter out low-quality instances and background images. Then, we develop a method to calculate feasible pasting locations on the image, ensuring that objects do not overlap with instances in the pasted area, which helps prevent the model from regenerating unwanted objects. Last, the instance is pasted onto a background image with a blending algorithm. The resulting dataset consists of 1 million image triplets.

Using the Masked-Region Guidance paradigm and the Syn4Removal dataset, we design a framework based on text-to-image diffusion models for object removal. To enhance the model's robustness to varying mask shapes from user input, we introduce a mask enhancement technique that simulates different mask shapes over removal targets. Additionally, we incorporate CLIP-based visual guidance to assist the model in better understanding the removal targets. The resulting model called *SmartEraser* outperforms previous methods significantly in both quantitative and qualitative evaluations. We choose the name *SmartEraser* because the model can "smartly" identify removal targets and remove it while preserving its surrounding region.

Our contributions are mainly summarized as follows:

- We propose Masked-Region Guidance, a novel paradigm for object removal, which effectively addresses issues of object regeneration and surrounding region distortion.
- We introduce Syn4Removal, a large-scale high-quality dataset for object removal, which contains over a million pairs of images across diverse scenes and object types.
- We carefully design a framework based on the text-to-image diffusion model by introducing mask enhancement and clip-based visual guidance.
- Extensive experiments demonstrate that SmartEraser significantly outperforms previous object removal methods in terms of quality and robustness.

## 2. Related Work

**Image inpainting.** Image inpainting aims to seamlessly fill missing regions in images by leveraging contextual information around the masked region. Based on the "mask-and-inpaint" paradigm, earlier works are mainly based on generative adversarial networks (GANs) [11, 31]. Typically, these methods [2, 4, 8, 24, 25, 27, 39, 43, 49] randomly mask part areas of images [19, 20, 28, 54] and then predict these masked regions. However, GAN-based approaches [12, 41, 42, 50, 51] often suffer from limited diversity and quality in generated content, resulting in blurry or unrealistic inpainted results.

Recently, approaches [1, 6, 9, 18, 30, 35, 47, 55] based on diffusion models have attracted significant attention due to their superior capability in generating image details. Among them, Repaint [30] injects the known region into intermediate images at all timesteps throughout the diffusion generation. SD-Inpaint [35] utilizes the mask and cropped image as input and fills the masked region through diffusion. While these approaches produce detailed textures, the generated content within the masked region often remains uncertain, with new objects occasionally appearing.

**Object removal.** The object removal task aims to eliminate specified objects from images based on user-provided masks, seamlessly repainting the removed area with unmasked content. For example, PowerPaint [55] uses learnable task embeddings from context-aware inpainting as the positive prompt and those from text-guided inpainting as the negative prompt to guide object removal. A recent work CLIPAway [9] leverages AlphaCLIP [38] and IP-Adapter [46] to inject background information into the model via cross-attention, ensuring removal results coherent with the background context. The common characteristic of these methods is that they retain the traditional "mask-and-inpaint" paradigm by discarding the masked regions of input images. In this paper, we claim that this paradigm is not suitable for the object removal task exactly due to the missing context of the masked region, causing incomplete removal and inconsistent context in removal results.

We also observe that some instruction-based methods [3, 48] attempt to remove objects using text prompts instead of the masked region. While they use the original image as input, their models are trained with removal results generated by inpainting models, which limits their performance. We believe that our synthetic strategy for building removal datasets could enhance the performance of these models.

There are several works [23, 36, 45, 48] aiming to construct datasets for object removal. For instance, GQA-Inpaint [48] and DEFACTO [23] synthesize ground truths with the help of image inpainting models. However, we find that relying on image inpainting leads to the performance of object removal being highly limited by the effect of inpainting. Another two works RORD [36] and ObjectDrop [45] acquire real-world triplets through photography, but the high collection costs result in fewer than $3.5k$ unique scenes. To incorporate with the Masked-Region Guidance, we construct a large-scale, high-quality dataset called Syn4Removal through a set of carefully designed strategies, which includes about 1 million triplets featuring diverse scenes and authentic ground-truth backgrounds with carefully filtering and pasting strategies.

# 3. SmartEraser

The goal of object removal is to completely erase the target object specified by a masked region and seamlessly generate consistent content based on the surrounding context. Most existing methods follow a "mask-and-inpaint" paradigm, where the masked area is excluded from the input, and the model attempts to inpaint the missing content based on the unmasked regions. While these methods can successfully remove objects in simple scenarios, they often regenerate new objects or produce blurs and artifacts in the masked region in more complex scenes.

In contrast, we propose a novel paradigm: *Masked-Region Guidance*. Instead of discarding masked regions, we retain it in the input, leveraging it to guide the removal process. This approach is based on the insight that, by knowing the exact removal targets, the model can precisely erase them while leaving the surrounding context maintained. In the following sections, we will first introduce the method of Masked-Region Guidance in Sec. 3.1. Then, we present Syn4Removal, a large-scale synthetic dataset designed for this paradigm in Sec. 3.2. Finally, we describe *SmartEraser*, a fine-tuned diffusion model based on the Mask-Region Guidance paradigm and Syn4Removal dataset in Sec. 3.3.

## 3.1. Masked-Region Guidance

Given an input image $\boldsymbol{x}$, and user-specified mask $\boldsymbol{m}$ containing the removal targets, the object removal task aims to produce the removal result $\hat{\boldsymbol{x}}$ while maintaining the realism of the original image. The mask $\boldsymbol{m}$ is a binary mask where 1 indicates the region containing the removal targets. Existing models [9, 35, 39, 55] are mostly based on the "mask-and-inpaint" paradigm, *i.e.*, utilizing the unmasked region $\boldsymbol{x} \odot (\mathbf{1} - \boldsymbol{m})$ and mask $\boldsymbol{m}$ to predict the results without removal object in the masked region. These models are learned to maximize the following objective:

$$\mathcal{L}' = P(\hat{\boldsymbol{x}} | \boldsymbol{m}, \boldsymbol{x} \odot (\mathbf{1} - \boldsymbol{m}); \Theta'), \tag{1}$$

where the conditional probability $P$ is modeled using a neural network with parameters $\Theta'$, $\odot$ specifies the element-wise multiplication.

In contrast, we introduce *Masked-Region Guidance*, a fundamentally different approach. Instead of discarding the masked region, we retain it in the input, allowing it to guide the removal process. Specifically, we replace the condition input from $[\boldsymbol{m}, \boldsymbol{x} \odot (\mathbf{1} - \boldsymbol{m})]$ to $[\boldsymbol{m}, \boldsymbol{x}]$. The mask $m$ is still applied to instruct the region for removal targets, but $\boldsymbol{x}$ now includes the masked region $\boldsymbol{x} \odot \boldsymbol{m}$ besides $\boldsymbol{x} \odot (\mathbf{1} - \boldsymbol{m})$. By including the masked region, we provide the model with valuable guidance during the object removal task. *First*, retaining the masked region allows the model to know exactly which objects need to be removed, reducing the risk of regenerating the object in the masked area. This helps the model focus on learning how to reconstruct the scene without the object. *Second*, the masked region is typically larger than the removal targets. This allows the model to copy the surrounding context of target objects in the masked region to the final result, significantly reducing the difficulty

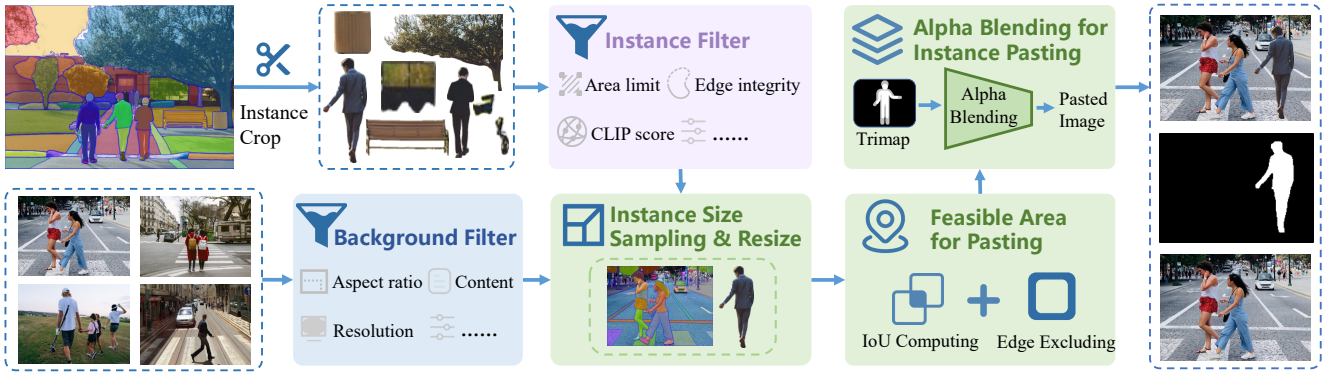



Figure 3. The data generation pipeline of Syn4Removal. We apply instances and images to construct a triplet consisting of an input image with removal targets, a mask, and the ground truth.

of synthesizing a larger area. Moreover, if the masked region contains complex textures, these can be retained, leading to more realistic results. In our Masked-Region Guidance paradigm, object removal is learned to maximize the objective:

$$\mathcal{L} = P(\hat{\boldsymbol{x}}|\boldsymbol{m}, \boldsymbol{x}; \Theta), \tag{2}$$

where the model is with parameters $\Theta$.

### 3.2. Syn4Removal

Now we have a new paradigm for training object removal models, but it cannot be directly applied using existing data construction methods. Typically, training data is created by masking parts of an image and having the model predict the masked content. However, this could lead the model to simply copy the masked content if applied to our paradigm. To overcome this, we require a dataset with triplets (input image, mask, and ground truth). However, existing real image datasets [36, 45] are limited in terms of scene variety, as they require photos taken before and after placing objects. Additionally, some synthetic datasets [23, 48] rely on inpainting models to generate pseudo-removed results, which don't fully capture the real removal process.

To address this, we propose a synthetic method for constructing object removal training data. Our approach involves pasting object instances from various images onto different backgrounds to form the input images. The mask of the pasted object serves as the input mask, and the original background image serves as the ground truth. Since the ground truth is the real image, it helps the model better capture the true distribution of the image, leading to more realistic object removal results. As a result, we constructed Syn4Removal, a large-scale high-quality data set that contains 1 million samples. The data construction pipeline is shown in Figure 3. Next, we will introduce each component of the overall pipeline.

**Instance Crop & Filter.** We directly use a public instance segmentation dataset [22] to get instances. We use the mask to get each instance from the image. While cutting instances, we also compute the corresponding CLIP similarity score with their classes and normalized area ratio in the original images. Then we perform the filter process, First, we exclude instances with areas larger than 95% or smaller than 5% to prevent potential low-quality segmentation results. Then, based on the previously calculated CLIP scores, we remove instances with low semantic relevance in each class. Given the variability in CLIP score distribution across different classes, we set individual CLIP score thresholds for each class according to previous work [53], filtering out instances with scores below these thresholds.

**Background Filter.** For background images, we use public datasets COCONut [7] and SAM-1B [21], along with segmentation annotations. We filter out low-resolution images, those with extreme aspect ratios, or those with a large number of segmented instances. The selected backgrounds are then combined to form 1 million high-quality images (for more details, see Supplementary).

**Instance Size Sampling & Resize.** To match the size of the instance with that of the background, we first calculate the average $\mu_c$ and variance $\sigma_c^2$ of the normalized area ratio for each class $c$. When pasting an instance onto a background, we sample scale $s$ from a Gaussian distribution $\mathcal{N}(\mu_c, \sigma_c^2)$, and scale the instance to $sHW$, where $H$ and $W$ are the height and width of the background image, respectively.

**Feasible Area for Pasting.** To ensure that the pasted instance does not cover existing instances in the background image, we calculate IoU between the pasted instance region and each instance in the background image, the feasible region based on IoU results is formulated as:

$$\mathbf{R}_1 = \{(x, y) \,|\, \text{IoU}(P_{x,y}, I_i) < r,\ I_i \in [1, I_{max}]\}, \tag{3}$$

where $P_{x,y}$ is the instance region if centered on $(x, y)$, $r$ is a IoU threshold, $I_{max}$ is the number of existing instances. Besides, to ensure the integrity of pasted instances, edge areas of background images are avoided, formulated as:

$$\mathbf{R}_2 = \{(x, y) | x \in [e_x, W - e_x], y \in [e_y, H - e_y]\}, \tag{4}$$

where $e_x$ and $e_y$ are half of the width and height of the bounding box of the instance mask, respectively. We take

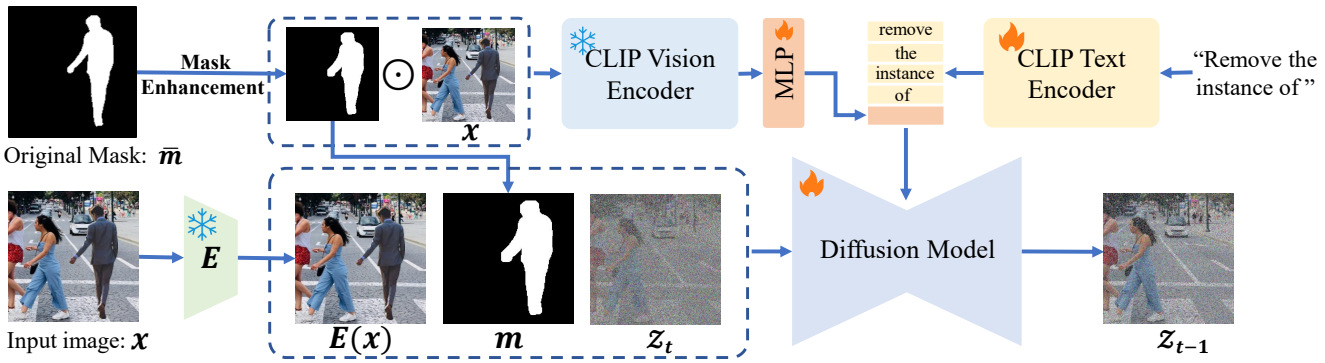



Figure 4. Overall framework of SmartEraser. It fully utilizes the *Masked-Region Guidance* paradigm and the Syn4Removal dataset.

the intersection region of $\mathbf{R}_1$ and $\mathbf{R}_2$, the final set of feasible locations is as follows:

$$\mathbf{R}_\mathrm{f} = \mathbf{R}_1 \cap \mathbf{R}_2. \tag{5}$$

We then randomly select a location from $\mathbf{R}_\mathrm{f}$ as the center point for pasting the instance. Based on the mask of the instance and location, we can get the instance mask centered on the location $\bar{\boldsymbol{m}}$.

**Alpha Blending for Instance Pasting.** To further ensure that the pasted instance blends harmoniously with the background, we apply alpha blending [13]. This method interpolates between the foreground instance and the background image, formulated as follows:

$$\boldsymbol{x} = \boldsymbol{\alpha} \odot \boldsymbol{x_i} + (1 - \boldsymbol{\alpha}) \odot \boldsymbol{x_b}, \tag{6}$$

where $\boldsymbol{x}$ is final constructed image, $\boldsymbol{x_i}$ is the instance, $\boldsymbol{x_b}$ is the background image, and $\boldsymbol{\alpha}$ is alpha blending value. $\boldsymbol{\alpha}$ is calculated through an alpha blending algorithm [13] that processes the mask $\bar{\boldsymbol{m}}$ to a trimap and then takes the trimap as input, which consists of three regions, *i.e.*, the foreground and background, and an excessively uncertain boundary region. So we have the triplet for object removal training: $\{\boldsymbol{x}, \bar{\boldsymbol{m}}, \boldsymbol{x_b}\}$, where $\boldsymbol{x}$ is input image with target object, $\bar{\boldsymbol{m}}$ is original mask, and $\boldsymbol{x_b}$ as the ground truth. Using this data generation pipeline, we finally get about 1 million triplets that form the Syn4Removal dataset.

### 3.3. Framework

Based on the proposed Masked-Region Guidance paradigm and the Syn4Removal dataset, we design a framework based on the text-to-image stable diffusion model for object removal. In this framework, we carefully design a user-friendly mask enhancement strategy and introduce CLIP-based visual guidance to further improve the capability of the model for object removal, as shown in Figure 4.

**Mask Enhancement**. In real-world scenarios, users usually provide a loose or tight mask around the object to be removed. If the model were trained only with precise object masks, there would be a significant gap in mask shape and size between training and inference. To address this,

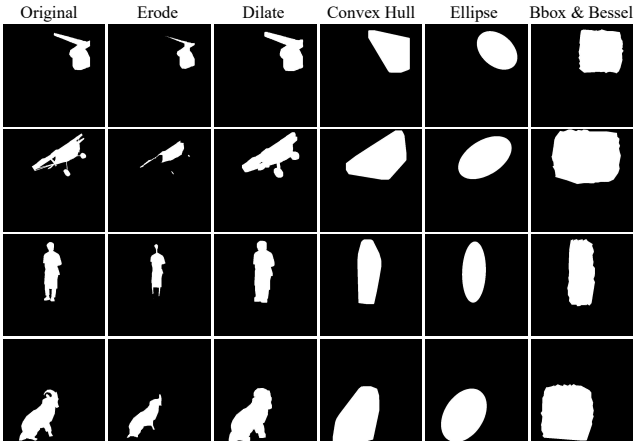



Figure 5. Mask shapes from different mask enhancement methods.

during training, we apply various mask deformation methods to simulate the user input mask shapes, as shown in Figure 5. These techniques help the model generalize to different mask forms. Specifically, we use six mask types to augment the object mask: (1) *Original mask*: this mask precisely outlines the target object. (2) *Eroded mask*: created by applying morphological erosion to the original mask, simulating scenarios where the user's mask may not fully cover the object. (3) *Dilated mask*: created by applying morphological dilation to the original mask. (4) *Convex hull mask*: constructed by calculating the convex hull that fully encompasses the original mask, then expanding it slightly. (5) *Ellipse mask*: generated by finding the smallest enclosing ellipse around the original mask, followed by a slight expansion. (6) *Bbox & Bessel mask*: a bounding box is generated to completely cover the original mask first, then bessel curves are constructed along each edge of the box to introduce an irregular boundary. So the overall mask enhancement process is illustrated as follows:

$$\boldsymbol{m} = \mathrm{ME}_i(\bar{\boldsymbol{m}}), \quad i \in \{1, ..., 6\}, \tag{7}$$

where $\boldsymbol{m}$ is the enhanced mask used during training, $\mathrm{ME}_i(\cdot)$ represents six different mask enhancement methods.

**CLIP-Based Visual Guidance**. Our SmartEraser is designed based on text-to-image diffusion model, the text

Figure 6. Qualitative comparison of SmartEraser and state-of-the-art methods. The samples are sourced from RORD-Val, which includes the ground truth after removal.

prompts play an important role in guiding the generation process. We aim to design a prompt that indicates the model what to remove during the generation process. Concretely, we first segment the region corresponding to the removal target in the image using its corresponding enhanced mask. This segmented region is then fed into the pre-trained CLIP [33] vision encoder to extract its visual feature(*e.g.*, class token). This feature is not precisely aligned with the text encoder output space. So we then apply a trainable MLP network to map the visual feature into the feature space of the text encoder. The mapped embedding is then appended to the tokens of the CLIP text embedding of the prompt "*Remove the instance of*". During the training process, the vision encoder is frozen, and the MLP and text encoder are trainable. So the CLIP-based visual guidance is formulated as follows:

$$\widehat{\boldsymbol{c}} = [\boldsymbol{\tau}_\theta(\boldsymbol{y}), \text{MLP}(\boldsymbol{\nu}_\theta(\boldsymbol{x} \odot \boldsymbol{m}))], \tag{8}$$

where $\boldsymbol{\tau}_\theta(\cdot)$ is text encoder, $\boldsymbol{\nu}_\theta(\cdot)$ is vision encoder, and $\boldsymbol{y}$ represents input text prompt "*remove the instance of*".

**Loss Function.** Suppose the $E(\cdot)$ is the VAE encoder, so the latent feature of input image $\boldsymbol{x}$ and its ground truth $\boldsymbol{x_b}$ is $\bar{\boldsymbol{z}} = E(x)$ and $\boldsymbol{z} = E(x_b)$, respectively, and $\boldsymbol{z}_t$ is the noisy feature of $\boldsymbol{z}$ with added noise $\epsilon$ at timestep $t$. So the overall training loss following the standard diffusion process, which is formulated as follows:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{x_b},t,\boldsymbol{\epsilon}} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, \bar{\boldsymbol{z}}, \boldsymbol{m}, \widehat{\mathbf{c}}, t)\|_2^2, \tag{9}$$

where $\boldsymbol{\epsilon}_\theta(\cdot)$ is the diffusion model.

# 4. Experiments

## 4.1. Settings

**Baselines.** We compare SmartEraser with methods in both image inpainting and object removal. For image inpainting, we evaluate against both GAN-based approaches (ZITS++ [4], MAT [24], LaMa [39], BLD [1]) and diffusion-based methods (RePaint [30], and SD-Inpaint [35]). Additionally, we compare it with specialized object removal frameworks, including CLIPAway [9] and PowerPaint [55], which represent the previous state-of-the-art performance on object removal benchmarks.

**Implementation details.** To enable a fair comparison, we follow previous experimental settings [9, 55] and finetune SmartEraser on the widely-used SD v1.5 model [35]. The training process involves fine-tuning both the UNet and text encoder components within the diffusion model and MLP for mapping visual features to text features space. SmartEraser is trained on the proposed Syn4Removal dataset with a batch size of 32, using the AdamW [29] optimizer for 500$k$ iterations and a learning rate of 2e-5 across all trainable modules.

**Evaluation benchmarks.** We evaluate the object removal capability of all baseline methods and our approach across three benchmarks in diverse scenarios. The benchmarks are constructed as follows:

- RORD-Val: Due to there are many triplets share the same background images as the ground truth in validation set of the RORD [36]. To ensure unique scenes and high object quality, we filter this subset to obtain RORD-Val, a evaluation benchmark of real-photography image pairs.
- DEFACTO-Val: Derived from the splicing section of DEFACTO [23], we exclude examples where intented targets for removal significantly overlap with instances in the background images.
- Syn4Removal-Val: Generated using the pipeline of Syn4Removal with MSCOCO [26] validation set as background images and OpenImages-v7 [22] validation set offering instances.

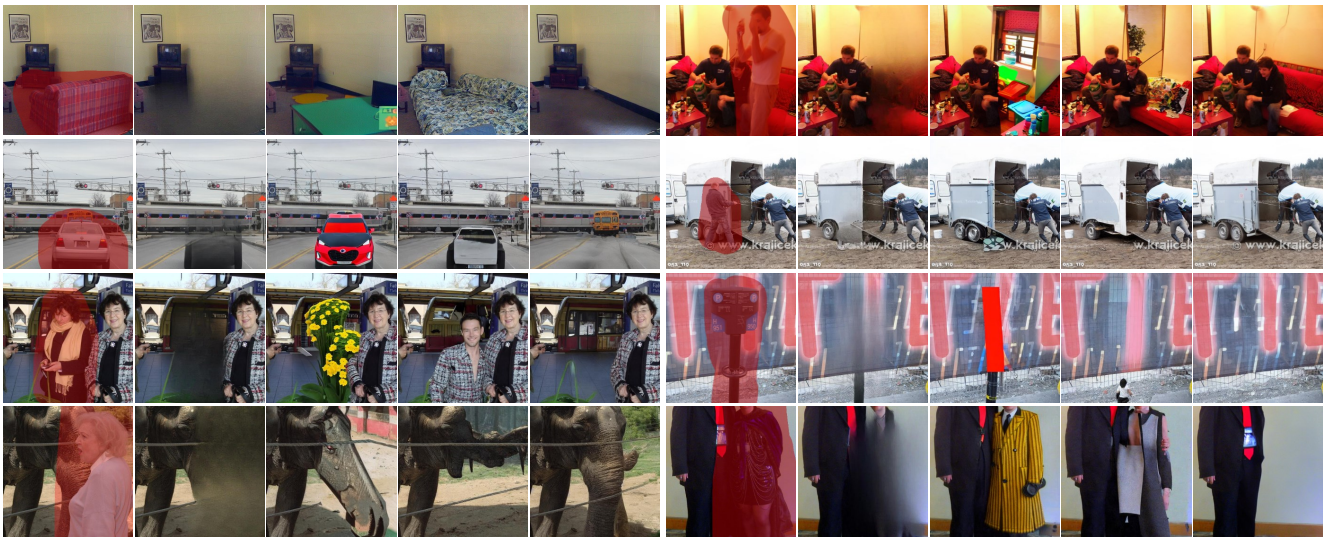



Figure 7. Qualitative comparison of SmartEraser and other methods in real-world user cases, in which users provide various mask shapes. The samples are sourced from real-world images and the validation set of MSCOCO [26].

Each benchmark comprises approximately 5k samples. During inference, following [9], we apply slight mask dilation to mitigate downscaling artifacts in SD-Inpaint-based methods [9, 35, 55].

**Evaluation metrics.** To quantitatively assess the performance of the object removal models, we consider three key aspects: (1) *overall image quality*, which is evaluated by Fréchet Inception Distance (FID) [14] and CLIP Maximum Mean Discrepancy (CMMD) [17]. (2) *consistency between the predicted region and the background context*, evaluated by REMOVE [5] metric. (3) *consistency between the predicted region and corresponding region in the ground truth*, assessed by LPIPS [52], SSIM [44], and PSNR [16].

### 4.2. Comparasion with Previous Methods

**Quantitative results.** We conduct extensive experiments to evaluate the performance of SmartEraser against baselines on the benchmarks. The results are shown in Tables 1, 2 and 3. SmartEraser significantly outperforms all other methods across all metrics on the three benchmarks. Notably, on the more complex RORD-Val dataset, which has larger masks and more challenging removal scenes, SmartEraser surpasses the previous state-of-the-art by margins of 10.3 in FID and 1.89dB in PSNR. This indicates the superior performance of SmartEraser for object removal. Besides, SmartEraser also achieves state-of-the-art performance on the simpler DEFACTO-Val and Syn4Removal-Val datasets.

**Qualitative results.** We present visualization comparison results of object removal using images from real scenes, shown in Figure 6. Baseline methods have limitations and may introduce unintended objects, create unrealistic structures, or produce artifacts within masked regions. In contrast, SmartEraser succeeds in effectively removing target objects while generating high-quality synthesized results, and excels in maintaining the contextual integrity and visual coherence of the scene.

| Method | FID ↓ | CMMD ↓ | ReMOVE ↑ | LPIPS ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|---|---|
| ZITS++ [4] | 53.440 | 0.838 | 0.820 | 0.388 | 0.522 | 15.442 |
| MAT [24] | 80.953 | 1.494 | 0.628 | 0.433 | 0.495 | 13.322 |
| LaMa [39] | 24.237 | 0.216 | 0.916 | 0.348 | 0.557 | 16.383 |
| BLD [1] | 73.086 | 1.760 | 0.834 | 0.471 | 0.467 | 15.775 |
| RePaint [30] | 71.542 | 1.639 | 0.812 | 0.381 | 0.510 | 15.861 |
| SD-Inpaint [35] | 69.502 | 0.324 | 0.857 | 0.369 | 0.537 | 16.111 |
| CLIPAway [9] | 25.458 | 0.123 | 0.915 | 0.333 | 0.577 | 17.434 |
| PowerPaint [55] | 24.058 | 0.294 | 0.926 | 0.308 | 0.602 | 18.100 |
| **SmartEraser** | **16.030** | **0.092** | **0.937** | **0.276** | **0.612** | **19.994** |

Table 1. Quantitative comparison of SmartEraser and previous methods on RORD-Val.

| Method | FID ↓ | CMMD ↓ | ReMOVE ↑ | LPIPS ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|---|---|
| ZITS++ [4] | 8.153 | 0.229 | 0.899 | 0.350 | 0.634 | 19.453 |
| MAT [24] | 15.257 | 0.321 | 0.742 | 0.407 | 0.575 | 14.541 |
| LaMa [39] | 6.601 | 0.201 | 0.926 | 0.351 | 0.683 | 21.735 |
| BLD [1] | 24.407 | 0.746 | 0.913 | 0.418 | 0.632 | 19.346 |
| RePaint [30] | 12.314 | 0.822 | 0.924 | 0.325 | 0.644 | 20.607 |
| SD-Inpaint [35] | 11.119 | 0.196 | 0.912 | 0.320 | 0.656 | 21.691 |
| CLIPAway [9] | 8.320 | 0.193 | 0.925 | 0.323 | 0.666 | 21.953 |
| PowerPaint [55] | 6.582 | 0.275 | 0.934 | 0.278 | 0.715 | 23.870 |
| **SmartEraser** | **3.405** | **0.106** | **0.939** | **0.257** | **0.734** | **25.363** |

Table 2. Quantitative comparison of SmartEraser and previous methods on DEFACTO-Val.

**User study.** To assess the visual quality subjectively, we conduct a user study with 30 participants. The study includes 20 image sets, each containing an original image, its mask, and removal results from four different methods, presented in random order. Participants are asked to select their preferred result based on three criteria: overall image quality, foreground-background consistency, and accuracy of object removal. Voting percentages in Table 4 show that

| Method | FID ↓ | CMMD ↓ | ReMOVE ↑ | LPIPS ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|---|---|
| ZITS++ [4] | 5.932 | 0.193 | 0.810 | 0.358 | 0.581 | 17.315 |
| MAT [24] | 19.263 | 0.300 | 0.700 | 0.374 | 0.570 | 13.357 |
| LaMa [39] | 5.246 | 0.097 | 0.932 | 0.325 | 0.606 | 19.319 |
| BLD [1] | 20.286 | 0.705 | 0.896 | 0.464 | 0.525 | 16.539 |
| RePaint [30] | 12.677 | 0.696 | 0.908 | 0.378 | 0.553 | 17.689 |
| SD-Inpaint [35] | 11.528 | 0.107 | 0.897 | 0.348 | 0.572 | 18.105 |
| CLIPAway [9] | 5.530 | 0.093 | 0.930 | 0.344 | 0.593 | 18.561 |
| PowerPaint [55] | 5.025 | 0.200 | 0.931 | 0.291 | 0.663 | 20.594 |
| **SmartEraser** | **4.386** | **0.053** | **0.939** | **0.269** | **0.672** | **22.029** |

Table 3. Quantitative comparison of SmartEraser and previous methods on Syn4Removal-Val.

| Method | Image Quality ↑ | FB-Consistency ↑ | Removal Accuracy ↑ |
|---|---|---|---|
| LaMa [39] | 0.10 | 0.11 | 0.21 |
| CLIPAway [9] | 0.18 | 0.12 | 0.11 |
| PowerPaint [55] | 0.16 | 0.12 | 0.10 |
| **SmartEraser** | **0.56** | **0.65** | **0.58** |

Table 4. Comparison of user study results in terms of image quality, foreground-background consistency, and removal accuracy.

users consistently favor the results from SmartEraser.

**Real-world user cases.** To further evaluate the object removal capability of SmartEraser in real-world user scenarios, we conduct a comparison with other methods (LaMa [39], CLIPAway [9], PowerPaint [55]), using masks provided by users. Figure 7 shows the comparison results. We can observe that SmartEraser exhibits a remarkable ability to accurately identify and remove target objects while preserving surrounding contextual details, particularly in scenarios involving imperfect or approximate masks. In the sample in the bottom row on the left, SmartEraser accurately identifies and removes the *person* indicated by the mask, while maintaining the integrity of the *elephant* in the background. In contrast, alternative methods tend to repaint the entire masked area, which leads to outputs that lack contextual coherence and visual realism due to the loss of details of the *elephant*. The upper second sample on the right provides a challenging scenario, the user-provided mask aims to remove the *man* whose feet are partially obscured by *floating text*. The previous methods fail to maintain the integrity of the *floating text*. However, SmartEraser accurately removes the *man* while preserving the *floating text*. This highlights SmartEraser's ability to effectively differentiate between removal targets and background content, even in cases of partial occlusion.

### 4.3. Ablation studies

To evaluate the effectiveness of the key components in our framework, we perform ablation studies and report both quantitative and qualitative results. We progressively modify the baseline (fine-tuning SD v1.5 with the "mask-and-inpaint" paradigm) by adding our proposed techniques: RG for masked-region guidance, ME for mask enhancement, and VG for CLIP-based visual guidance. The results are displayed in Figure 8 and Table 5.

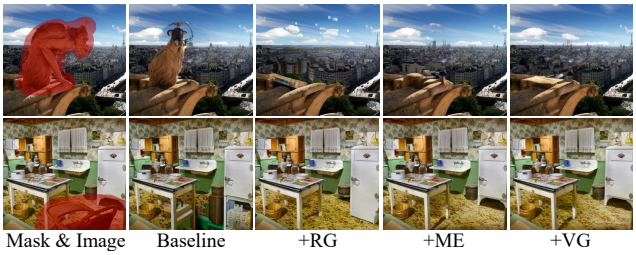



Figure 8. Qualitative ablation comparison of our method. From left to right, we progressively add each proposed component.

| Method | FID ↓ | CMMD ↓ | ReMOVE ↑ | LPIPS ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|---|---|
| Baseline | 33.131 | 0.245 | 0.908 | 0.355 | 0.544 | 16.535 |
| +RG | 24.526 | 0.121 | 0.923 | 0.339 | 0.562 | 17.176 |
| +ME | 19.421 | 0.108 | 0.931 | 0.301 | 0.591 | 18.917 |
| +VG | 16.030 | 0.092 | 0.937 | 0.276 | 0.616 | 19.994 |

Table 5. Quantitative ablated comparison on RORD-Val.

**Effect of the Masked-Region Guidance paradigm**: We observe that the baseline model often regenerates objects instead of properly removing them, as it is trained in a traditional "inpainting" manner. Incorporating masked-region guidance significantly improves the removal performance by reducing the risk of object regeneration. However, due to the mismatch between the mask shape used during training and the user-defined mask during testing, the model does not always perfectly remove the target objects and preserve the surrounding context.

**Contribution of Mask Enhancement**: By simulating real-world user mask shapes over object removal during training, we find that it helps reduce the gap between model training and testing using user input mask, significantly keeping the surrounding regions and further improving removal results.

**Value of CLIP-based Visual Guidance**: CLIP visual guidance aids the model in better understanding the removal targets, yielding the best performance on RORD-val across all metrics. This indicates that the visual guidance helps the model perform object removal tasks more effectively.

## 5. Conclusion

In conclusion, we present *SmartEraser*, a state-of-the-art object removal model built on the novel *Masked-Region Guidance* paradigm. SmartEraser outperforms existing methods by smartly identifying the target object to remove while effectively preserving the surrounding context. To facilitate research on this paradigm, we propose Syn4Removal, a large-scale, high-quality dataset containing over a million image triplets, specifically designed for object removal tasks. Through extensive experiments, we demonstrate that SmartEraser achieves superior performance in both quality and robustness compared to previous object removal methods.

## References


[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42 (4):1–11, 2023. 1, 3, 6, 7, 8

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 3

[4] Chenjie Cao, Qiaole Dong, and Yanwei Fu. Zits++: Image inpainting by improving the incremental transformer on structural priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 6, 7, 8

[5] Aditya Chandrasekar, Goirik Chakrabarty, Jai Bardhan, Ramya Hebbalaguppe, and Prathosh AP. Remove: A reference-free metric for object erasure. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 7901–7910, 2024. 7

[6] Pau de Jorge, Riccardo Volpi, Puneet K. Dakania, Philip H. S. Torr, and Rogez Gregory. Placing objects in context via inpainting for out-of-distribution segmentation. In *The European Conference on Computer Vision (ECCV)*, 2024. 3

[7] Xueqing Deng, Qihang Yu, Peng Wang, Xiaohui Shen, and Liang-Chieh Chen. Coconut: Modernizing coco segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21863–21873, 2024. 4

[8] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11358–11368, 2022. 2

[9] Yigit Ekin, Ahmet Burak Yildirim, Erdem Eren Caglar, Aykut Erdem, Erkut Erdem, and Aysegul Dundar. Clipaway: Harmonizing focused embeddings for removing objects via diffusion models, 2024. 1, 3, 6, 7, 8

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1

[11] 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, 2

[12] Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14134–14143, 2021. 2

[13] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR 2011*, pages 2049–2056, 2011. 5

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 7

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

[16] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 7

[17] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315, 2024. 7

[18] 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, 2024. 3

[19] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 4

[22] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020. 4, 6

[23] Jens Lehmann, Daniel Gerber, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. Defacto-deep fact validation. In *International semantic web conference*, pages 312–327. Springer, 2012. 2, 3, 4, 6

[24] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10758–10768, 2022. 2, 6, 7, 8

[25] Xiaoguang Li, Qing Guo, Di Lin, Ping Li, Wei Feng, and Song Wnag. Misf: Multi-level interactive siamese filtering for high-fidelity image inpainting. *CVPR*, 2022. 2

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6, 7

[27] Qiankun Liu, Zhentao Tan, Dongdong Chen, Qi Chu, Xiyang Dai, Yinpeng Chen, Mengchen Liu, Lu Yuan, and Nenghai Yu. Reduce information loss in transformers for pluralistic

image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11347–11357, 2022. 2

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 2

[29] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[30] 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, 3, 6, 7, 8

[31] Mehdi Mirza. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[33] 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

[34] 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. 1

[35] 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 (CVPR)*, pages 10684–10695, 2022. 1, 3, 6, 7, 8

[36] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *BMVC*, page 542, 2022. 2, 3, 4, 6

[37] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1

[38] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13019–13029, 2024. 3

[39] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 1, 2, 3, 6, 7, 8

[40] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1

[41] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4692–4701, 2021. 2

[42] Wentao Wang, Jianfu Zhang, Li Niu, Haoyu Ling, Xue Yang, and Liqing Zhang. Parallel multi-resolution fusion network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14559–14568, 2021. 2

[43] Wentao Wang, Li Niu, Jianfu Zhang, Xue Yang, and Liqing Zhang. Dual-path image inpainting with auxiliary gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11421–11430, 2022. 2

[44] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 7

[45] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. *arXiv preprint arXiv:2403.18818*, 2024. 2, 3, 4

[46] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3

[47] Yifu Chen, Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Zhineng Chen and Tao Mei. Improving text-guided object inpainting with semantic pre-inpainting. *ECCV*, 2024. 3

[48] Ahmet Burak Yildirim, Vedat Baday, Erkut Erdem, Aykut Erdem, and Aysegul Dundar. Inst-inpaint: Instructing to remove objects with diffusion models. *arXiv preprint arXiv:2304.03246*, 2023. 2, 3, 4

[49] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 2

[50] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14114–14123, 2021. 2

[51] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M. Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14164–14173, 2021. 2

[52] 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. 7

[53] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-

paste for instance segmentation using clip and stablediffusion. In *International Conference on Machine Learning*, pages 42098–42109. PMLR, 2023. 4

[54] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[55] 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, 2023. 1, 3, 6, 7, 8