A. Data Generation

In Sec. 3.2, we employed pretrained model checkpoints and implementations from the Hugging Face diffusers library [51]. Specifically, for text-to-image generation, we used Stable Diffusion 2.0 ("stabilityai/stable-diffusion-2-base") with a DDIM scheduler, and SDXL Turbo ("stabilityai/sdxl-turbo"). For text-based image inpainting, we utilized the SD 2.0 inpainting model ("stabilityai/stable-diffusion-2-inpainting"). Furthermore, our 1,024 seeds range from 0 to 1,023 inclusive, and we use `torch.Generator("cuda").manual_seed(seed)` to assign the seed used by the model.

A.1. Synthetic Prompts for Image Composition Analysis

We create a set of 880 prompts by pairing 40 object categories with 22 modifiers in the format "a [modifier] [object category]". These modifiers include 21 adjectives and the empty string.

- **Adjectives:** big, small, red, blue, pale, dark, transparent, shiny, dull, rustic, smooth, rough, bright, muted, round, simple, elegant, antique, monochrome, intricate, sleek

- **Object categories:** bicycle, car, motorcycle, airplane, bus, truck, boat, fire hydrant, bench, bird, cat, dog, horse, sheep, cow, elephant, zebra, giraffe, backpack, umbrella, suitcase, sports ball, skateboard, surfboard, tennis racket, fork, knife, spoon, bowl, apple, pizza, donut, cake, chair, couch, laptop, cell phone, clock, vase, teddy bear

A.2. Dataset for Inpainting Applications

We curated 500 pairs of images and inpainting masks for object removal and object completion applications, as described in Sec. 3.2. In particular, for the object removal use case, we employed images and annotations from the Open Images dataset [20, 22], and we used "clear background" as the text prompt. To create the inpainting mask, we dilated the instance segmentation mask to ensure coverage of the object. Additionally, for the object completion use case, we sampled images from the MS-COCO dataset [25] and used InstaOrder [23] to determine occlusion relationships to create inpainting masks. We used the category of the object to complete as the text prompt.

A.3. Licenses for Existing Datasets

The MS-COCO dataset [25] and the PartiPrompts benchmark [57] are under a CC BY 4.0 license. For the Open Images dataset [20, 22], the images are under a CC BY 2.0 license and the annotations are under a CC BY 4.0 license.

B. Classifier for Predicting Seed Number

We trained a lightweight transformer, EfficientFormer-L3 [24], to predict the seed used to generate an image. For our 1,024-way classification task, we utilized 9,000 training, 1,000 validation, and 1,000 test images per seed as mentioned in Sec. 3.3. The prompts for these images are dense captions by LLaVA 1.5 [26]. Moreover, we set a batch size of 128 and train for six epochs, which obtains a model checkpoint with over 99.9% validation and test accuracy. Our classifier uses the AdamW optimizer [27] with learning rate 0.0002 and weight decay 0.05. We apply data augmentations during training, which include resizing each image to have a shorter edge of size 224 using bicubic interpolation, center cropping the image to size 224×224, and randomly flipping the image horizontally with probability 0.5. During validation and testing, we only resize and center crop the images.

C. Compute Resources

To generate our dataset in Sec. 3.2, we utilized 32 A100 GPUs for roughly 24 days. Additionally, all the experiments in Sec. 3.3, 3.4, and 4 were performed on an RTX 4090 GPU with 24GB of memory. One of the longest experiments was training the classifier to predict seed number in Sec. 3.3, which took at most three days.

D. Additional Qualitative Results

We provide extra visualizations of the Grad-CAM from our classifier that predicts seed number in Figure 13 of the supplemental. We also show more examples of seeds that often produce a 'border' around the image in Figure 14 of the supplemental. Moreover, we present additional examples of good seeds and seeds that generate "text artifacts" for object removal and completion applications in Figures 15 and 16 of the supplemental, respectively.

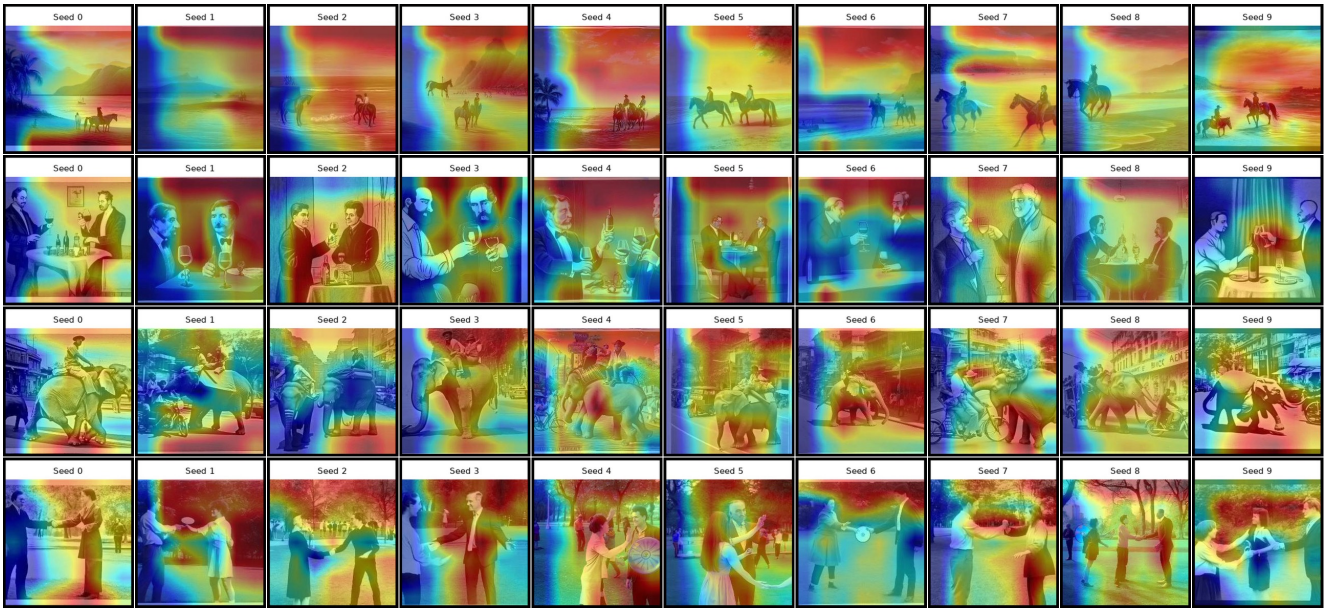

Figure 13. Additional Grad-CAM [13, 42] visualizations for our classifier trained to predict the seed number for an image. We note that it is difficult to interpret what makes seeds easily distinguishable by looking at these visualizations.

Figure 14. Additional examples of seeds that tend to generate a 'border' near the image boundaries.

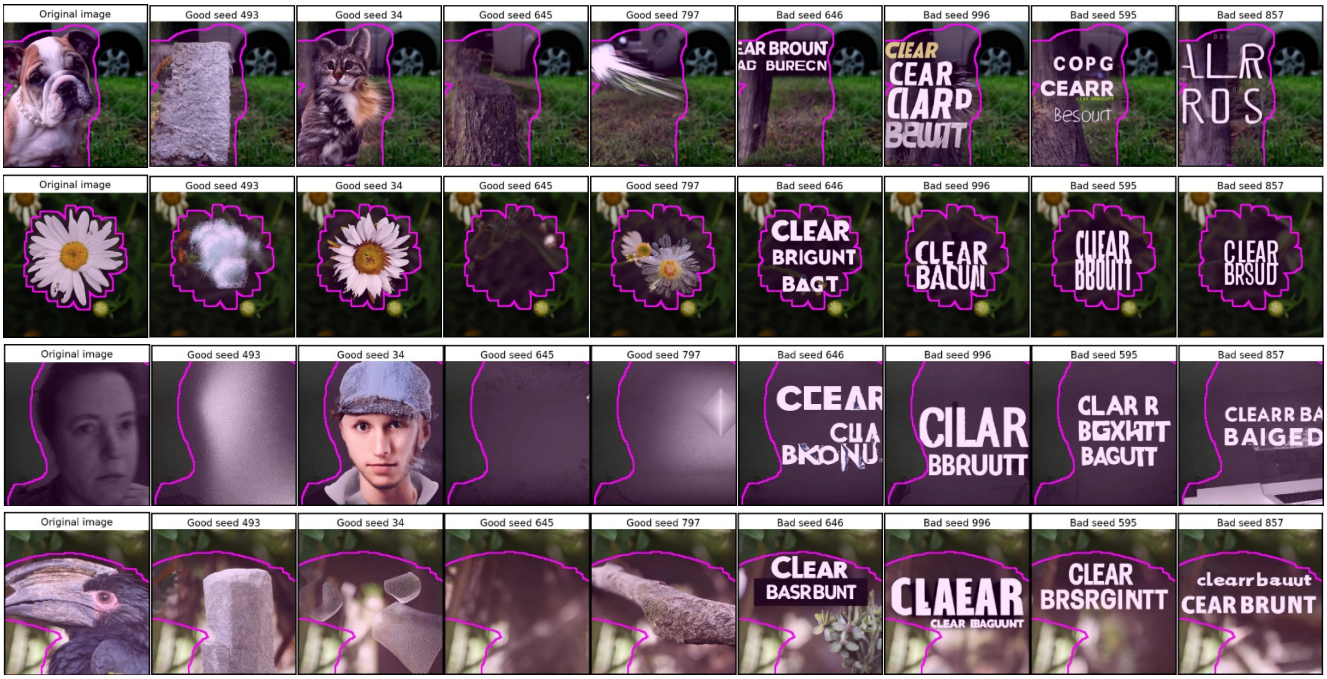

Figure 15. Additional examples of the four best seeds and four worst seeds in terms of how much unwanted text artifacts are inserted during object removal.

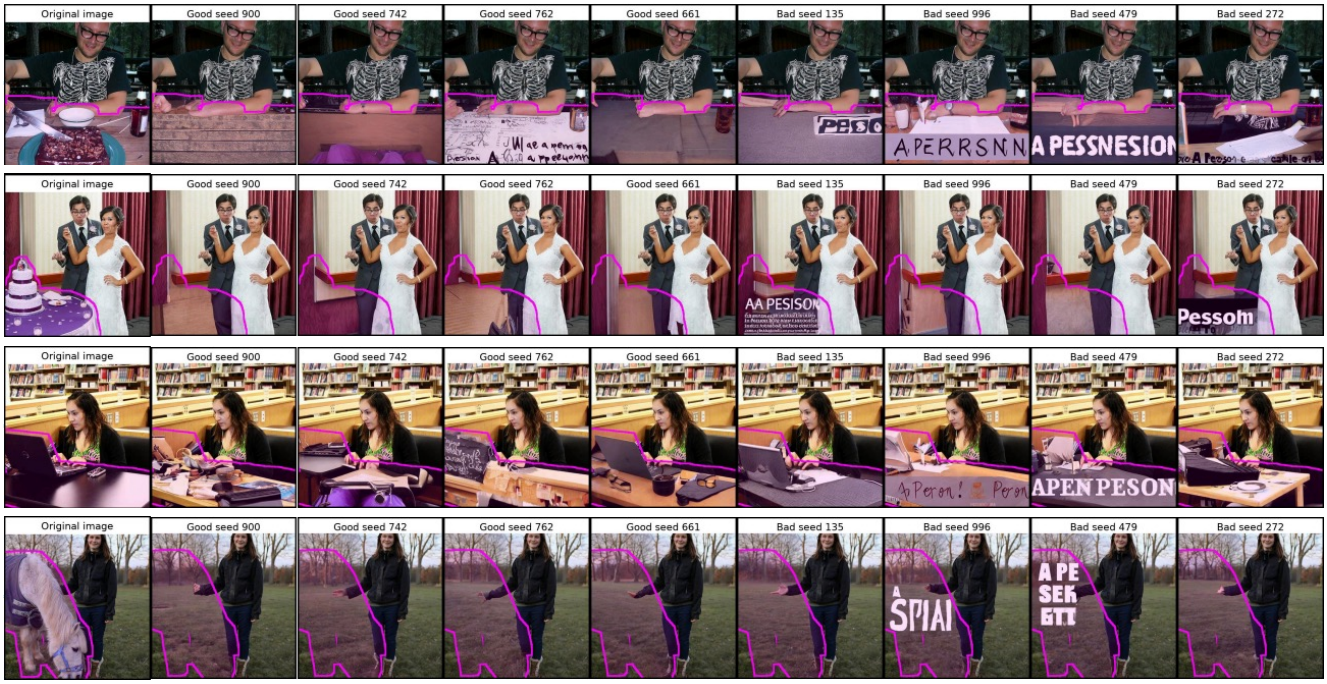

Figure 16. Additional examples of the four best seeds and four worst seeds in terms of how much unwanted text artifacts are inserted during object completion.