Secret Seeds in Text-to-Image Diffusion Models

Katherine Xu
University of Pennsylvania

Lingzhi Zhang
Adobe Inc.

Jianbo Shi
University of Pennsylvania

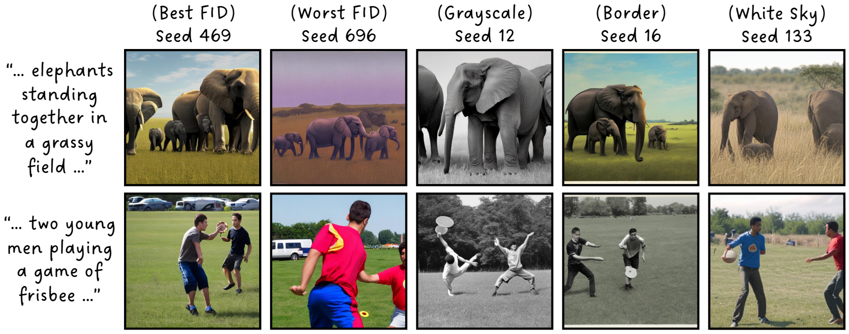

(Best FID)
Seed 469
(Worst FID)
Seed 696
(Grayscale)
Seed 12
(Border)
Seed 16
(White Sky)
Seed 133
"... elephants standing together in a grassy field ..."
"... two young men playing a game of frisbee ..."

Figure 1: We reveal that the seed number impacts various visual elements in text-to-image generation.

Abstract

Recent text-to-image diffusion models have facilitated creative and photorealistic image synthesis. By varying the random seed, we can generate many images for a fixed text prompt. The seed controls the initial noise and, in multi-step diffusion inference, the noise used for reparameterization at intermediate timesteps in the reverse diffusion process. However, the impact of the seed on the generated images remains relatively unexplored. We conduct a scientific study into the influence of seeds during diffusion inference on interpretable visual dimensions and, moreover, demonstrate improved image generation. Our analyses highlight the importance of selecting good seeds and offer practical utility for image generation.

1 Introduction

Text-to-Image (T2I) diffusion models [2, 3, 5, 22, 24, 25, 39] have advanced image synthesis significantly, enabling the creation of photorealistic, high-resolution images. However, their training requires substantial computational resources, limiting such research to a few well-equipped labs. Despite these limitations, many studies have enhanced image generation during inference by feature re-weighting [30], gradient-based guidance [8, 29, 34], or integration with multimodal LLMs [4, 38].

We propose an inference technique to enhance image generation by exploring 'secret seeds' in the reverse diffusion process. Inspired by research like Torch.manual_seed(3407) [23], which revealed that well-chosen neural network initialization seeds can outperform poorly chosen ones in image classification, we investigate whether 'golden' or 'inferior' seeds similarly impact image quality in T2I diffusion inference. Using the pretrained T2I model Stable Diffusion (SD) 2.0 [25] across 1,024 seeds, we discovered that the best 'golden' seed achieved an FID [12, 27] of **21.60**, whereas the worst 'inferior' seed only reached an FID of **31.97**—a significant difference within the community. This finding sparked our curiosity to understand several scientific questions: What does the seed control in T2I diffusion inference? Can seeds be distinguished by the images they generate? Do they control interpretable image dimensions, and if so, how can this be leveraged to enhance image generation?

2 Understanding Diffusion Seeds

2.1 What do seeds control in the reverse diffusion process?

Random seeds play different roles in deep learning depending on the context. During deep network training, they often influence the initialization of neural network weights, data scheduling, augmentation strategies, and stochastic regularization techniques such as dropout [33]. We aim to understand what the seeds control in the reverse diffusion process and during diffusion inference.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

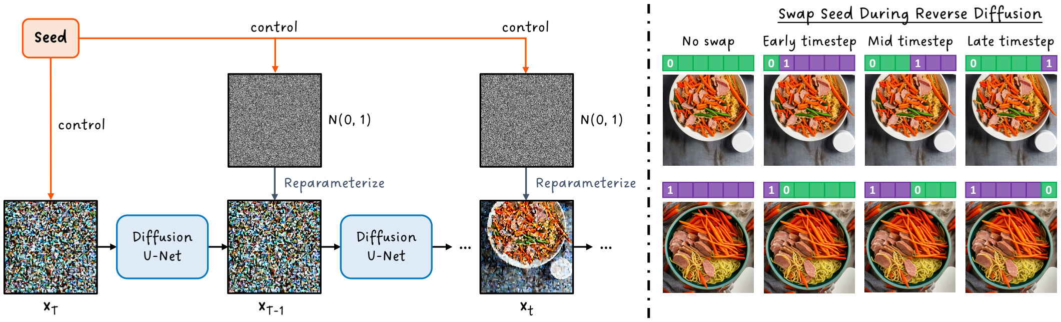


Figure 2: **Left:** Overview of how the seed controls the initial noise x_T and intermediate x_t via the sampled noise in multi-step diffusion inference. **Right:** We swap the seed number at early, mid, and late timesteps of the reverse diffusion process, showing an example with seeds 0 and 1. Interestingly, the seed mostly influences the initial noisy latent, rather than intermediate timesteps.

We focus on latent diffusion models as described by Rombach et al. [25], although the same principles apply to pixel diffusion models. Theoretically, in the traditional multi-step reverse diffusion process, both the initial noisy latent variables and the noise used for reparameterization [14] at each timestep are sampled from a Gaussian distribution, introducing randomness. We visualize this process on the left side of Figure 2. At the implementation level, we confirmed that random seeds are used as inputs to compute these variables [36]. In a distilled one-step diffusion model, such as SDXL Turbo [26], the random seeds only determine the initial noisy latent, as there are no intermediate denoising steps.

In multi-step diffusion inference, seeds determine both the initial latent variables and the reparameterization noise at each timestep. To understand the separate impacts of the initial latent configuration and the reparameterization step on the generated images, we conducted a simple "seed swap" study shown on the right side of Figure 2 using the DDIM scheduler [32] with 40 inference steps. In our study, we first set the seed to i and begin the reverse diffusion process. Then, at an intermediate timestep, we change the seed to j and complete the image generation process. We explore using seeds 0 and 1 for both i and j, as well as swapping the seed at early, mid, and late timesteps of the reverse diffusion process. Despite these variations, we found that the initial noisy latent significantly controls the generated content, while the random noise introduced at intermediate reparameterization steps has no visible impact on the generated images, as shown on the right side of Figure 2.

2.2 Data Generation

To conduct a large-scale seed analysis, we gather prompts for text-to-image generation that capture a broad spectrum of natural visual content. We sample 20,000 images from the MS-COCO 2017 train set [19] and generate dense captions using LLaVA 1.5 [20]. For each prompt, we sample 1,024 seeds ranging from 0 to 1,023 and generate images using two models: SD 2.0 [25] and SDXL Turbo [26].

2.3 How discriminative are seeds based on their generated images?

We train a 1,024-way classifier to predict the seed number used to produce a given image, employing 9,000 training, 1,000 validation, and 1,000 test images per seed. Remarkably, seeds are highly differentiable based on their images. After only six epochs, our classifier trained on images from SD 2.0 [25] achieved a test accuracy of 99.99%, and the classifier trained on images from SDXL Turbo [26] reached a test accuracy of 99.96%. However, it is unclear what makes seeds discernible, as the Grad-CAM [11, 28] visualization in Figure 3 is not easily interpretable. These findings suggest that seeds may encode unique visual features, prompting us to explore their impact across several interpretable dimensions.

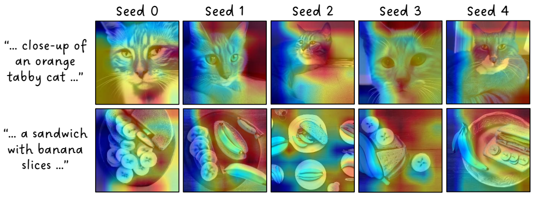

Seed 0
Seed 1
Seed 2
Seed 3
Seed 4
"... close-up of an orange tabby cat ..."
"... a sandwich with banana slices ..."

Figure 3: Grad-CAM [11, 28] of our classifier trained to predict the seed used to create an image.

2.4 Impact of Seeds on Interpretable Visual Dimensions

Image Quality. To assess the image quality related to each of 1,024 seeds, we chose 10,000 prompts and their corresponding generated images, and then computed the FID score [12, 27] against 10,000 real MS-COCO images [19]. Surprisingly, we observed a major difference in FID between the best and worst seeds. The 'golden' seed 469 for SD 2.0 achieved a low FID of 21.60, while the 'inferior' seed 696 scored 31.97—a disparity considered significant within the community.

Next, we determine whether the seed rankings are generalizable across prompts. In Figure 4, we plot the ranked seeds for FID using images from SD 2.0 and SDXL Turbo generated by distinct sets of

10,000 prompts, and we reveal a high degree of overlap between the seed patterns. This consistency underpins our proposed enhancements to inference strategies detailed in Sections 3.1 and 3.2.

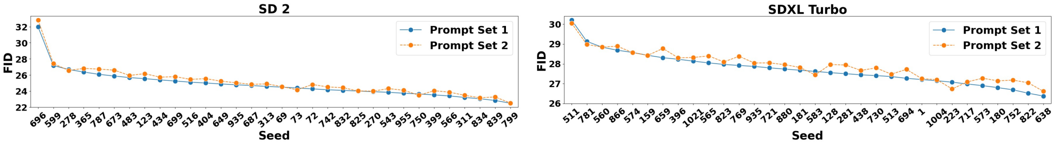


Figure 4: We sort seeds by FID [12] using 10,000 images in Prompt Set 1, and then display the FID for the same seeds using another 10,000 images in Prompt Set 2. Lower FID indicates better quality.

Image Style. We study whether specific seeds produce unique style patterns across prompts. Drawing on established methods in image texture and style transfer [9, 10], we compute style representations by extracting the Gram matrix — which measures pairwise cosine similarity across channels — from a pretrained deep network [31] at multiple layers. After reshaping the Gram matrix and reducing its dimensionality [1, 35], we have a compact 2D vector for each image that captures its style. For $N = 1024$ seeds and P prompts, this results in a feature dimension of $N \times (2 \times P)$, combining the style representation across the generated images for each seed. We further reduce [1, 35] this aggregated style representation per seed from $N \times (2 \times P)$ to $N \times 2$. Finally, a subset of seeds are visualized in Figure 5, providing a clear visual representation of style clustering at the seed level.

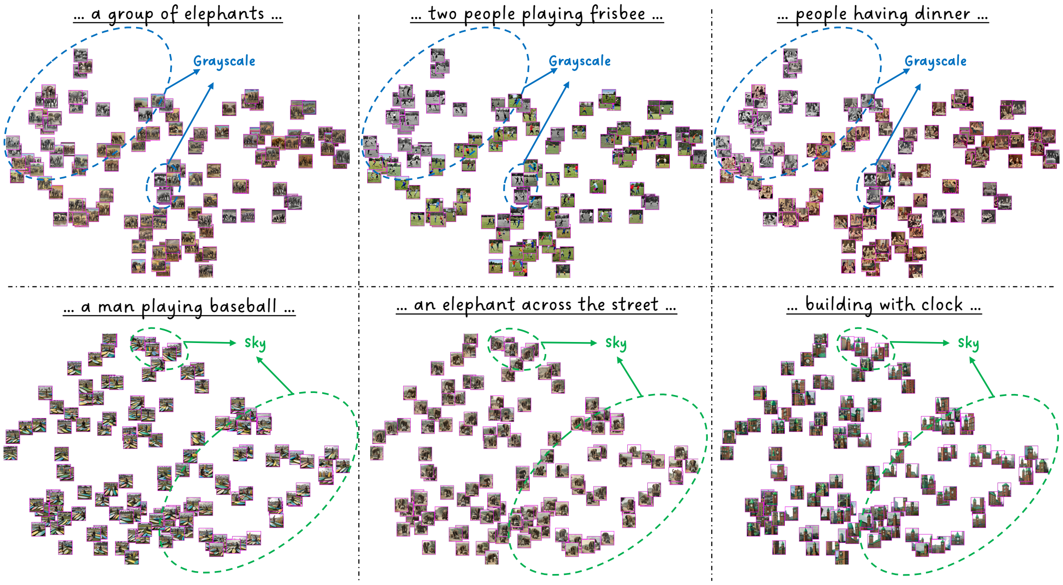


Figure 5: Style embedding clustering across various prompts, with each position corresponding to a unique seed. Certain seeds tend to generate grayscale images for SD 2.0 (top), while others frequently produce images with 'white sky' regions for SDXL Turbo (bottom). **Please zoom-in to check.**

In Figure 5, certain seed groups consistently generate grayscale images irrespective of the prompt, and some seeds tend to produce images with prominent sky regions. Furthermore, in Figure 6, a select group of seeds often generates images with a 'border' near the edges. These findings demonstrate that individual seeds exhibit distinct tendencies in style generation across prompts.

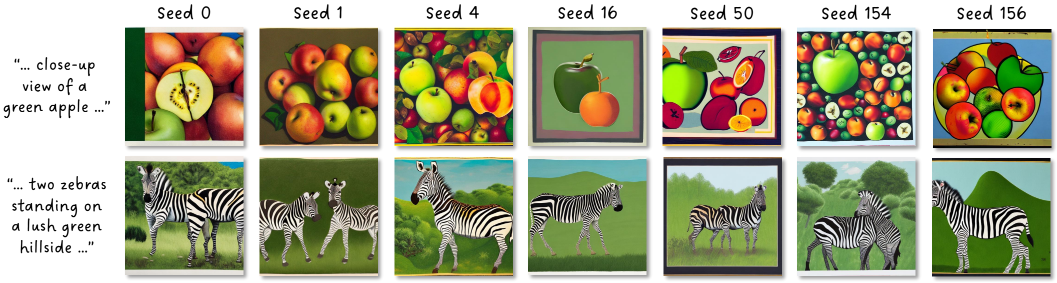

Seed 0
Seed 1
Seed 4
Seed 16
Seed 50
Seed 154
Seed 156
"... close-up view of a green apple ..."
"... two zebras standing on a lush green hillside ..."

Figure 6: We observe that certain seeds produce a "border" around the image for SD 2.0.

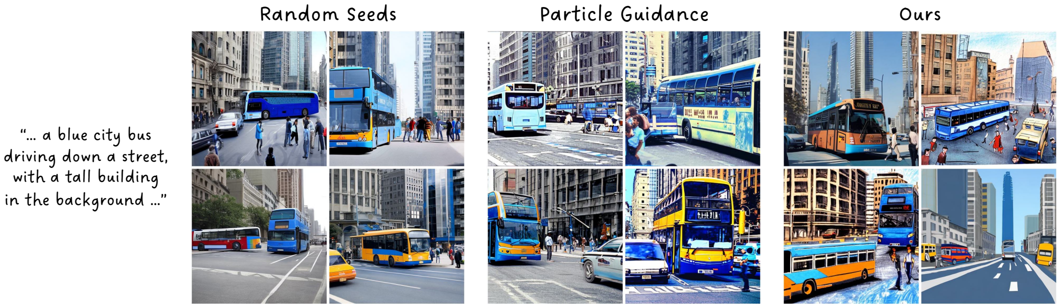

Random Seeds
Particle Guidance
Ours
"... a blue city bus driving down a street, with a tall building in the background ..."

Figure 7: We show that simply generating images using "diverse" seeds can promote style variation.

3 Practical Applications

3.1 High-Fidelity Inference

In Section 2.4, we observed that 'golden' seeds tend to generate images with significantly better quality. This inspires us to think—how much can we improve the image quality compared to random generations by simply leveraging these 'golden' seeds? Specifically, we identified 65 'golden' seeds for SD 2.0 that excel in image quality and evaluated their performance relative to random seeds by generating images using SD 2.0 with a different set of 10,000 prompts. Our 'golden' seeds achieved a lower FID score of 19.05 ± 0.06 compared to 19.33 ± 0.21 with random seeds across three trials.

3.2 Controllable Diversity in Style

A typical image generation interface presents the user with four samples per prompt. Moreover, prior methods encourage the diversity of generated images using primarily gradient-based methods, such as Particle Guidance [7]. In Section 2.4, our results highlight that the seed has a strong influence on image style. Thus, can we obtain more diverse images in style by merely sampling 'diverse' seeds?

To select $C = 4$ diverse seeds, we represent each seed by a feature vector $\mathbf{f}$ capturing its style, as discussed in Section 2.4. We then employ farthest point sampling using these features. We randomly pick the first seed $s_0 \sim \mathcal{U}\{0, 1023\}$ and iteratively select the next three seeds to maximize the distance in feature space from the already selected seeds, where S is our set of diverse seeds.

$$s_i = \arg\max_{s \notin S} \min_{s' \in S} \|\mathbf{f}(s) - \mathbf{f}(s')\|, \quad \text{for } i = 1, \ldots, C-1 \tag{1}$$

To evaluate whether our well-chosen seeds improve diversity over random seeds and Particle Guidance [7], we calculate the similarity between the C images synthesized from a different set of $P = 500$ prompts. We measure the pairwise cosine similarity of image features and average the similarity scores across prompts. Intuitively, a lower pairwise similarity score means higher diversity.

$$\text{Style Similarity} = \frac{1}{P} \sum_{i=1}^{P} \left(\frac{1}{\binom{C}{2}} \sum_{j=1}^{C} \sum_{k=j+1}^{C} \cos(\mathbf{f}_{ij}, \mathbf{f}_{ik}) \right) \tag{2}$$

In Table 1, we observe that our diverse seeds outperform random seeds and Particle Guidance [7] in generating images with varying styles. Additionally, we show visual comparisons in Figure 7.

Table 1: We compare the diversity in style of images generated using our diverse seeds, Particle Guidance [7], and random seeds. Lower style similarity scores indicate more diverse generations. We display the mean and standard deviation based on three trials.

	Style Similarity for SD 2.0 (↓)	Style Similarity for SDXL Turbo (↓)
Random Seeds	0.981 ± 0.001	0.993 ± 0.000
Particle Guidance	0.980 ± 0.000	—
Our Diverse Seeds	**0.970 ± 0.000**	**0.984 ± 0.000**

4 Conclusion

In this work, we investigated the role of "random" seeds in the reverse diffusion process, exploring their differentiability based on generated images and their impact on interpretable visual dimensions. Notably, our 1,024-way classifier trained to predict the seed number for a generated image achieved over 99.9% test accuracy in just a few epochs. Encouraged by this finding, we identified 'golden' seeds that produce images with better visual quality and discovered that certain seeds create 'grayscale' images or add borders. Our analyses aid in enhancing image synthesis during inference without significant computational overhead by merely sampling these special seeds.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[4] Bin Cao, Jianhao Yuan, Yexin Liu, Jian Li, Shuyang Sun, Jing Liu, and Bo Zhao. Synartifact: Classifying and alleviating artifacts in synthetic images via vision-language model. *arXiv preprint arXiv:2402.18068*, 2024.

[5] 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.

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[7] Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-iid diverse sampling with diffusion models. *arXiv preprint arXiv:2310.13102*, 2023.

[8] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023.

[9] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Neural Information Processing Systems*, 2015.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[11] Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[12] 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. *Advances in neural information processing systems*, 30, 2017.

[13] Samuel Hoffstaetter and contributors. Python tesseract. `https://github.com/h/pytesseract`, 2014.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[17] Hyunmin Lee and Jaesik Park. Instance-wise occlusion and depth orders in natural scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21210–21221, 2022.

[18] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

[19] 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.

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[22] Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Wuerstchen: An efficient architecture for large-scale text-to-image diffusion models, 2023.

[23] David Picard. Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*, 2021.

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[26] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[27] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.3.0.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[29] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023.

[30] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv preprint arXiv:2309.11497*, 2023.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[34] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[36] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[37] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[38] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. *arXiv preprint arXiv:2401.11708*, 2024.

[39] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

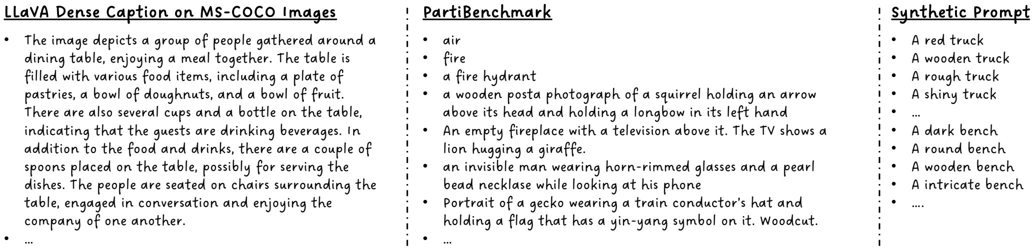
A Data Generation

In Section 2.2, we employed pretrained model checkpoints and implementations from the Hugging Face diffusers library [36]. We used Stable Diffusion 2.0 ("stabilityai/stable-diffusion-2-base") with a DDIM scheduler, and SDXL Turbo ("stabilityai/sdxl-turbo"). 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. Figure 8 showcases our various text prompts.

Additionally, in Section C, we investigate whether seeds influence the image layout, such as the main subject's scale, location, and depth. To enable a more controlled scientific study, 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

Lastly, going beyond text-to-image applications, our studies on image inpainting in Section D reveal that some seeds consistently generate 'text artifacts' instead of completing pixels, indicating that one could improve inpainting quality by using seeds that minimize these artifacts. We curated 500 pairs of images and masks for object removal and object completion applications, where the mask typically covers an object in the original image. In particular, for the object removal use case, we employed images and annotations from the Open Images dataset [15, 16], 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 [19] and used InstaOrder [17] to determine occlusion relationships to create inpainting masks. We used the category of the object to complete as the text prompt. For these inpainting cases, we used the SD 2.0 inpainting model ("stabilityai/stable-diffusion-2-inpainting").



LLaVA Dense Caption on MS-COCO Images

- The image depicts a group of people gathered around a dining table, enjoying a meal together. The table is filled with various food items, including a plate of pastries, a bowl of doughnuts, and a bowl of fruit. There are also several cups and a bottle on the table, indicating that the guests are drinking beverages. In addition to the food and drinks, there are a couple of spoons placed on the table, possibly for serving the dishes. The people are seated on chairs surrounding the table, engaged in conversation and enjoying the company of one another.
- ...

PartiBenchmark

- air
- fire
- a fire hydrant
- a wooden posta photograph of a squirrel holding an arrow above its head and holding a longbow in its left hand
- An empty fireplace with a television above it. The TV shows a lion hugging a giraffe.
- an invisible man wearing horn-rimmed glasses and a pearl bead necklase while looking at his phone
- Portrait of a gecko wearing a train conductor's hat and holding a flag that has a yin-yang symbol on it. Woodcut.
- ...

Synthetic Prompt

- A red truck
- A wooden truck
- A rough truck
- A shiny truck
- ...
- A dark bench
- A round bench
- A wooden bench
- A intricate bench
- ....

Figure 8: A visualization of three different types of text prompts used in our study.

B Classifier for Predicting Seed Number

We trained a lightweight transformer, EfficientFormer-L3 [18], 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 Section 2.3. The prompts for these images are dense captions by LLaVA 1.5 [20]. 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 [21] 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 Image Composition

Moving beyond style, we examine whether seeds create distinctive image compositions, such as consistent object locations and sizes. We generate images using 880 synthetic prompts consisting of 40 object categories paired with 22 modifiers, which includes adjectives and the empty string. For each image, we segment [6] the object and compute an image composition feature vector that contains the object's centroid (x, y) coordinates, size, and depth [37] relative to the image. On the left side of Figure 9, we visualize the distribution of the object mask's centroid for the category "horse." Remarkably, the object's position stays relatively the same despite slight prompt alterations. On the right side of Figure 9, we observe an analogous pattern in the object's size and depth for the category "bowl." Overall, we observe that the location, size, and depth of generated objects are largely dependent on the specific seed used, consistent across the same object categories and irrespective of the text modifiers in the prompts.

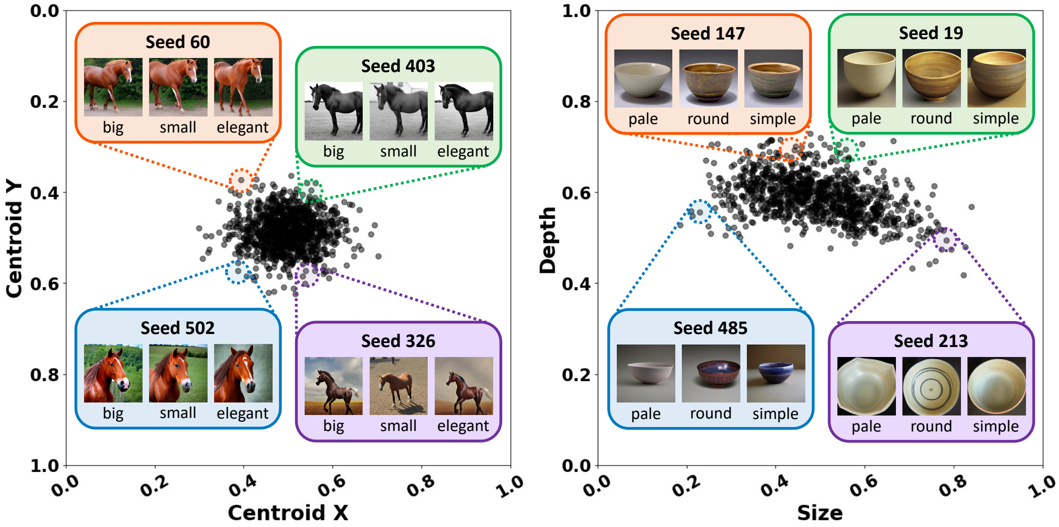


Figure 9: We observe that seeds produce images with unique and consistent compositions for a given object category. Each data point represents a seed. For each seed, we combine image composition features from 22 prompts with slight variations like "a pale bowl" and "a round bowl." Then, we apply dimensionality reduction [1, 35] for visualization. **Left:** Distribution of object centroid (x, y) coordinates. **Right:** Distribution of object depth and size relative to the image.

Controllable Diversity in Composition. Following our approach in Section 3.2, we explore whether we can generate more diverse images in composition by sampling 'diverse' seeds. We employ $P = 440$ prompts and $C = 4$ images per prompt, but it's important to note that if no objects are detected in an image, then the image is not used to compute similarity. In Table 2, we observe that our diverse seeds outperform random seeds and Particle Guidance [7] in generating images with varying compositions for SD 2.0. Interestingly, our well-chosen seeds aid in diversifying image composition for SD 2.0 but not for SDXL Turbo. We show visual comparisons in Figure 10.

Table 2: We compare the diversity in composition of images generated using our diverse seeds, Particle Guidance [7], and random seeds. Lower composition similarity scores indicate more diverse generations. We display the mean and standard deviation based on three trials.

	Composition Similarity for SD 2.0 (↓)	Composition Similarity for SDXL Turbo (↓)
Random Seeds	0.971 ± 0.001	0.988 ± 0.000
Particle Guidance	0.972 ± 0.000	—
Our Diverse Seeds	**0.961 ± 0.001**	0.988 ± 0.000

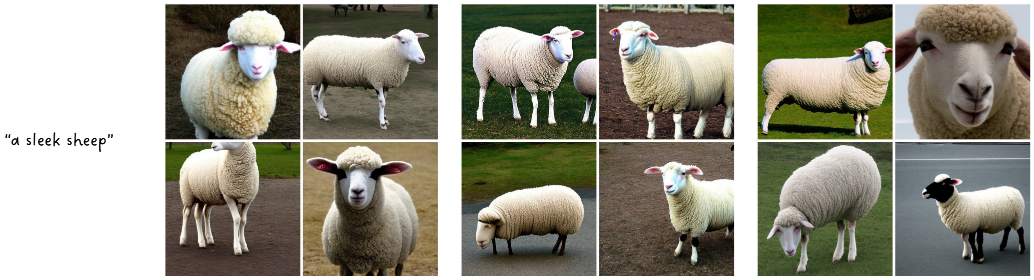


Figure 10: We show that simply generating images using "diverse" seeds can promote layout variation.

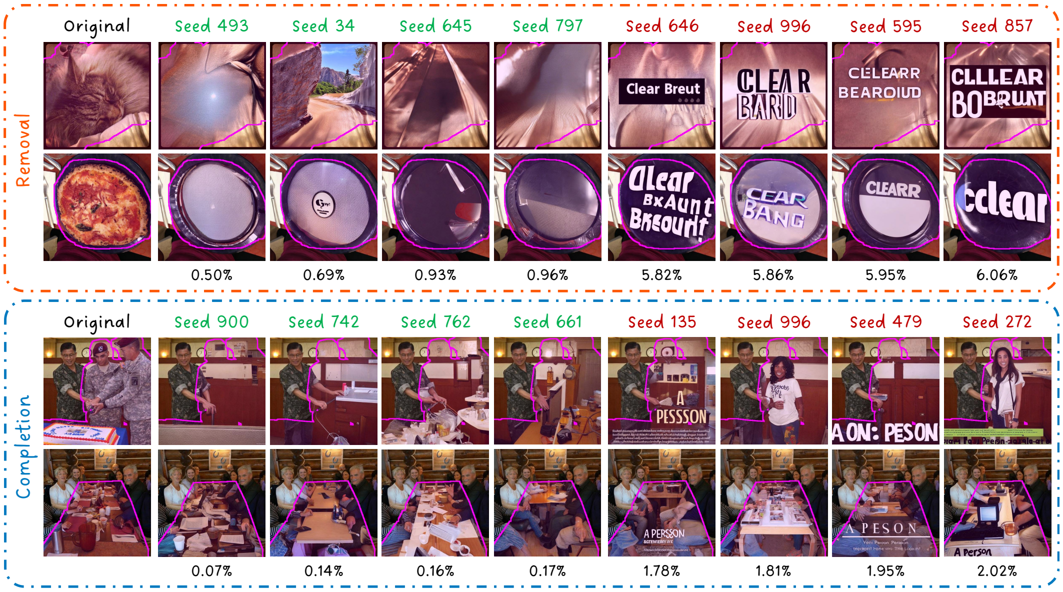


Figure 11: We discover that certain seeds tend to insert unwanted text within the inpainting region, outlined in pink. **Top:** We aim to remove the object using the prompt "clear background." **Bottom:** We attempt to complete the object using a prompt that specifies the object category.

D Improved Text-based Inpainting

In Sections 3.1 and 3.2, we demonstrated that carefully selecting the seed provides a straightforward, training-free approach to enhance the visual quality, human preference, and diversity of images generated by text-to-image diffusion models. But, the potential of image generation extends beyond text-to-image applications. This poses an intriguing question—can we also uncover 'golden' seeds for text-based image inpainting tasks, such as object removal and object completion?

We gathered 500 pairs of images and inpainting masks for the object removal and object completion applications. We employed the text prompt "clear background" for the removal case, and we used a prompt corresponding to the original object category for the completion case. We then generated images using a text-based diffusion inpainting model. We observed that some images contain unwanted text in the inpainting region that often mimics the prompt. To quantify the presence of text, we applied optical character recognition [13] and calculated the average proportion of text artifacts within the inpainting mask across all images from each seed. As illustrated in Figure 11, certain seeds are prone to inserting text in both removal and completion scenarios.

E Additional Qualitative Results

As illustrated in Figure 12, the top and bottom three seeds according to FID indeed reveal that the highest-rated seeds produce images that are more visually pleasing. Moreover, we show extra examples of seeds that often produce a 'border' around images in Figure 13, and we provide more visualizations of the Grad-CAM from our classifier that predicts seed number in Figure 14. Lastly, we present additional examples of good seeds and seeds that generate "text artifacts" for object removal and completion applications in Figures 15 and 16, respectively.

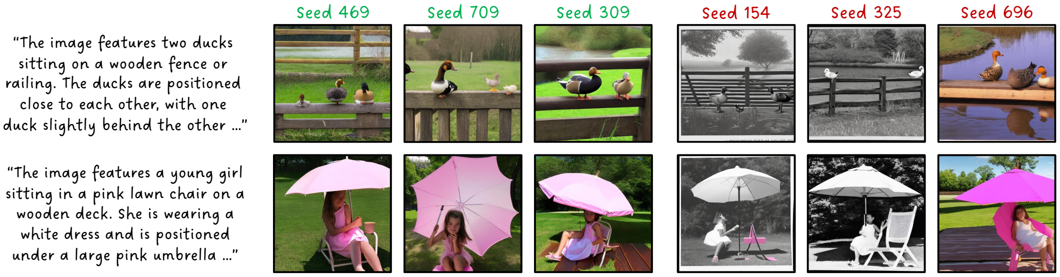

Seed 469
Seed 709
Seed 309
Seed 154
Seed 325
Seed 696
"The image features two ducks sitting on a wooden fence or railing. The ducks are positioned close to each other, with one duck slightly behind the other ..."
"The image features a young girl sitting in a pink lawn chair on a wooden deck. She is wearing a white dress and is positioned under a large pink umbrella ..."

Figure 12: We compare the top three best and worst seeds for SD 2.0 using FID [12].

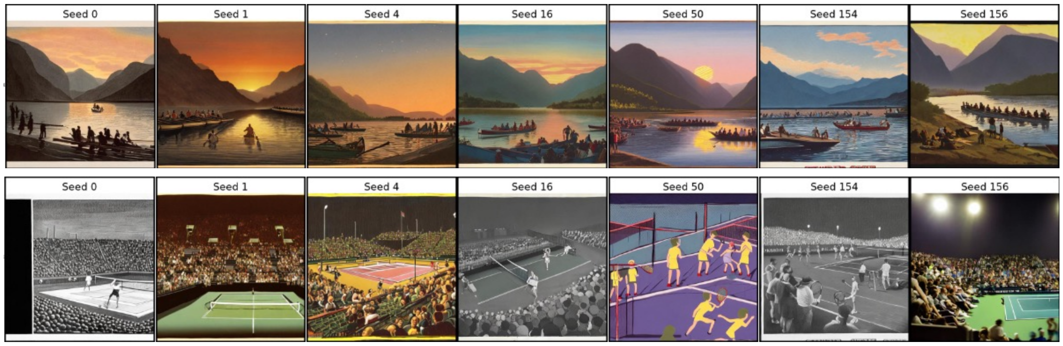


Figure 13: Additional examples of seeds that tend to generate a ‘border’ near the image boundaries.

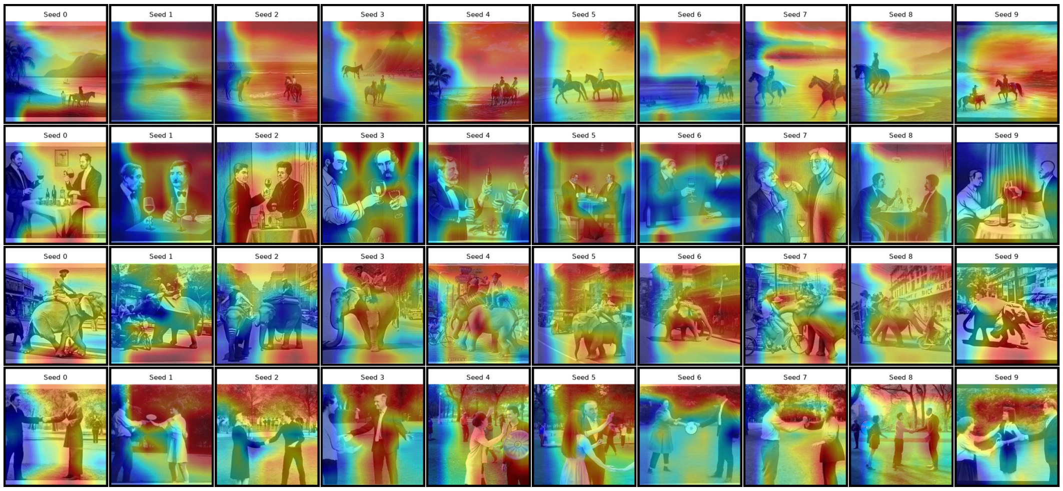


Figure 14: Additional Grad-CAM [11, 28] visualizations for our classifier trained to predict the seed number for an image. It is difficult to interpret what makes seeds easily distinguishable by looking at these visualizations, prompting us to study the impact of seeds across interpretable dimensions.

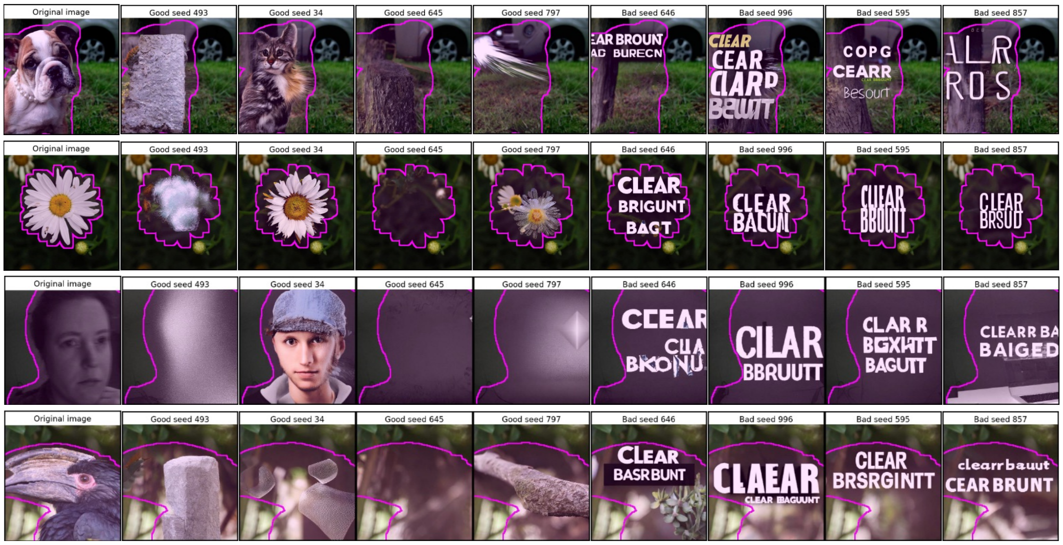


Figure 15: Additional examples of the four best seeds and four worst seeds in terms of how much unwanted text artifacts are inserted during object removal.

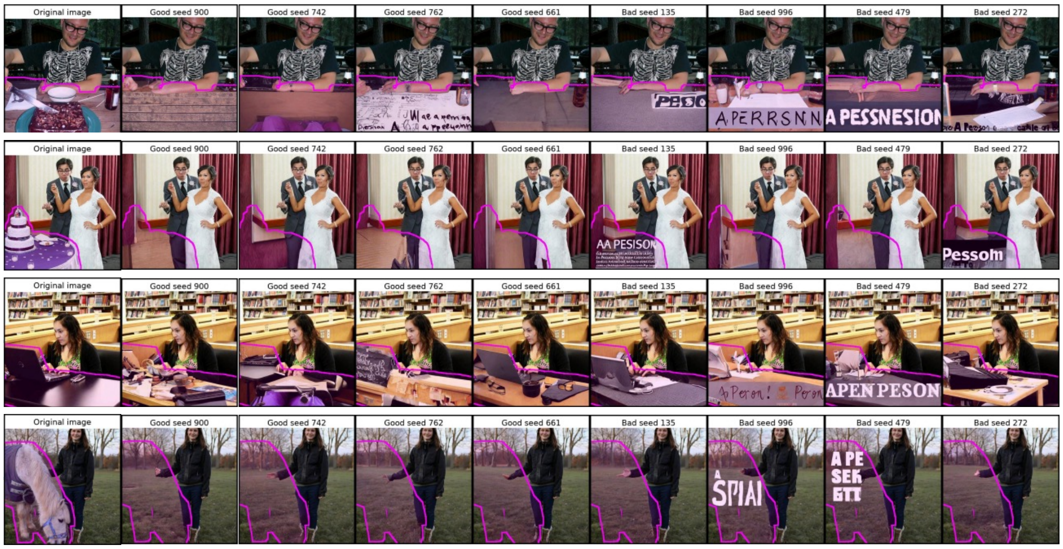


Figure 16: Additional examples of the four best seeds and four worst seeds in terms of how much unwanted text artifacts are inserted during object completion.