Improving Faithfulness of Text-to-Image Diffusion Models through Inference Intervention

Danfeng Guo[1], Sanchit Agarwal[2], Yu-Hsiang Lin[2], Jiun-Yu Kao[2]
Tagyoung Chung[2], Nanyun Peng[1], Mohit Bansal[3]
[1] University of California, Los Angeles
[2] Amazon AGI
[3] University of North Carolina, Chapel Hill

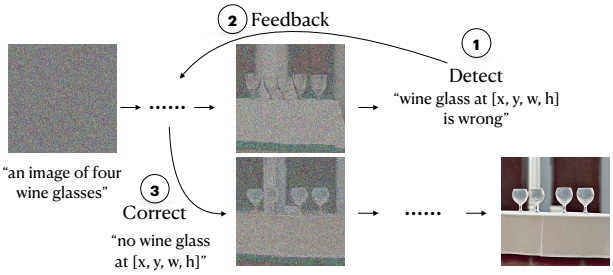
Abstract

Text-to-Image diffusion models have shown remarkable capabilities in generating high-quality images. However, current models often struggle to adhere to the complete set of conditions specified in the input text and return unfaithful generations. Existing works address this problem by either fine-tuning the base model or modifying the latent representations during the inference stage with gradient-based updates. Not only are these approaches computationally expensive, but also they usually only improve limited kinds of errors (e.g., the count of objects). In this work, we propose an intervention-based mechanism to enhance the faithfulness of diffusion models by controlling the denoising process. Starting with layout-conditional diffusion models, our approach first detects incorrectly-generated/missing objects during denoising steps. Next, a layout is constructed from the erroneous objects (feedback). Finally, we return to an earlier denoising step. The new layout is fed to the diffusion model to obtain its latent representation. Correction is applied by composing the new latents with the original ones and continuing the generation process, thereby driving the generation away from erroneous directions. As additional feedback and correction strategy, we also explore retrieval-augmented generation to help the model recover missing objects. We conduct experiments on VPEval and HRS-Bench datasets and measure faithfulness across four dimensions; presence of objects, object counts, scale of objects and spatial relations between objects. Compared to GLIGEN, the state-of-the-art model on the VPEval dataset, our approach significantly improves on all metrics (+6.7% average accuracy increase). On HRS-Bench dataset, it also outperforms existing models in count and scale metrics.

1. Introduction

Text-to-image (T2I) diffusion models have become one of the most popular image generation models nowadays.

(a) Examples of unfaithful generations from Dall-E 3. The images do not reflect the correct quantity/spatial relationship of objects.



(b) An overview of our method to improve faithfulness by error detection, feedback, and correction.

Figure 1. Examples of unfaithful generations and a brief overview of our method.

Given a piece of text that describes a scene, models like Imagen [34] and DALL·E [32] are able to generate beautiful images often as good as human artists. However, images generated by diffusion models often fail to faithfully reflect the full set of conditions specified in the text prompt. This unfaithfulness can manifest in various forms. For example, the generated image misses an object mentioned in the text input, or the number of objects in the generated image and the text prompt do not match, or the objects are generated in locations that contradicts the prompt specification. Figure 1a shows examples of unfaithful generations.

Enhancing the faithfulness of generated images involves incorporating a variety of conditions beyond mere text prompts. These can include layouts [22, 36], retrieved image examples [5], and segmentation maps [8]. Layout-based diffusion models have set the benchmark for faithful generation [7]. To better utilize the layout, prevalent strategies include training a fusion module that integrates bounding box information into local visual features [38] or designing a layout-based auxiliary loss function [12]. However, fine-tuning these models is time-consuming. To mitigate high cost of fine-tuning, recent studies have explored inference-time tactics to improve generation fidelity by controlling the denoising process. Most approaches in this direction focus on restricting the generated objects in specified bounding boxes by steering the latent representations from the diffusion model to a direction that minimizes their defined auxiliary loss [3, 19, 23, 35] (e.g., designing the loss as the difference between cross-attention values outside and inside the bounding boxes to restrict the objects in specified regions). However, such approaches often require repeated computation of gradients during the denoising steps, which is notably resource-intensive. Moreover, these methods typically resolve only a subset of issues because the loss functions are only designed for single purpose, such as to prevent blending of objects (e.g., a half-cat-half-dog animal) [23] or to fix object counts in the generated image [19].

In this paper, we propose a training-free interventive approach designed to rectify multiple types of generation errors within the denoising process, thus improving the faithfulness of generation. First, we employ an off-the-shelf layout generation model to create a layout, which serves as an auxiliary input for the diffusion model. Next, our intervention strategy (an overview shown in Figure 1b) unfolds in three steps:

1. During the denoising stage, we apply an object detector to the noisy, intermediate output.
2. The detection results are compared to the input layout and additional layouts, consisting of the erroneous objects are created as feedback. We also incorporate additional feedback utilizing a retrieval-based approach to supply reference images for missing objects.
3. The feedback layouts are fed to the model to generate their latent representations. These new representations are then composed with the latent representation of the original image via the composable diffusion method such that the generation is steered away from erroneous directions.

Overall, our mechanism has the following merits:

- It requires neither fine-tuning nor back-propagation and can efficiently correct errors that have been generated in early denoising steps.
- It can be used to correct several types of errors including missing objects, incorrect quantity, mislocated objects, and objects with wrong scale relationships.
- Our intervention-correction approach is more general; many new methods can fit in our three-step system.

We test on VPEVAL and HRS-Bench datasets, which directly measure the faithfulness of generation in multiple aspects (count, spatial relationship, scale, etc). Our method shows significant improvement over baselines on all metrics, and outperforms other denoising control algorithms on all VPEVAL metrics, HRS count, and HRS scale metrics.

2. Related Works

2.1. Faithful Generation with Diffusion Models

Diffusion models can perform conditional generation in a classifier-free manner [15]. Classifier-free diffusion models can generate images conditioned on arbitrary texts and images. Although existing models generate aesthetically pleasing images, they often do not accurately follow all the details in the prompt.

To address this problem, one popular method is to fine-tune the model with improved architecture or additional guidance. [12] applies an object detector during training and modifies the loss inside and outside of the detected bounding boxes separately. [5] retrieves reference images from its database to help diffusion model generate corresponding objects. [22, 30, 36] uses layouts which specify the locations of each object to help the generation. [13] uses segmentation maps to help localize the objects in the prompt. Besides, [21] designs a human feedback model to predict if the generations are matched with the prompts. Then train the diffusion model to maximize the human feedback score.

The main effort to control the generation process focuses on modifying the latents with the help of the image-text cross-attention maps extracted from the conditioning layers of diffusion models. For most works, a loss function is defined and computed from the cross-attention maps. Then, at each denoising step, the latents are updated using gradient descent that minimizes the loss. Works of this type include [3, 4, 10, 19, 35, 37]. [3] defines the loss as the negative of maximum cross-attention values across all patches. It helps enforce the diffusion models to generate the target objects. [19] define a counting loss, for which the number of objects is estimated from the cross-attention maps. This helps the model to generate correct quantity of objects. BoxDiff [35] defines the loss as the cross-attention values outside the bounding box minus those inside the bounding box, plus a corner constraint. It helps restrict the objects inside their bounding boxes. Although the aforementioned works improve the faithfulness, a limitation is that back-propagation is performed at each denoising step, which is

more time-consuming than forward propagation. This problem is worse when generating high-resolution images.

There are also other works trying to improve faithfulness of image generation. [26] merges the latents of the original generation and the reference image during the generation, such that the reference image is fused into the main image at specified locations. [23] applies an object detector on cross-attention maps and use the detected boxes to arrange the spatial locations of objects. This approach addresses the object mixing problem, while less helpful in some scenarios such as generating the correct number of objects. [20] generates multiple candidate images and uses an auto-selection method to select the one closest to the text prompt. [27] detect errors in the final image and re-initialize the noise in the erroneous regions. [14] checks errors during the inference step and rectifies the cross-attention map of diffusion U-Net. The recent Dall-E 3 [2] rewrites the text prompt with details to help the model understand the prompt better. However, it still fails to understand the location words such as "left" and "right" [2].

In conclusion, the strategies to improve the faithfulness is either resource-intensive or limited in the types of errors they can correct. In comparison, our method is training-free, backpropagation-free, and it can address a broader spectrum of generation errors. luated along various dimensions such as existence of objects, number of objects etc.

3. Methods

We follow common practice to start with layout information for improving the faithfulness of text-to-image generators. Layout encompasses both an object's category and its position within an image. Typically, a layout is represented as a sequence of labels accompanied by their bounding box coordinates for all the objects in the image. (e.g., {"*tennis racket* $< x_1, y_1, x_2, y_2 >$", "*woman* $< b_1, c_1, b_2, c_2 >$"}). The image generation task is then conditioned on both the text prompt and the layout. With layout as additional guidance, objects in the generated image are more likely to reside in their respective bounding boxes. Experiments in [7] show that layout-based generation constructs images with higher fidelity compared to text-based generation. To generate layouts from text prompt, we first apply the layout generation model in [7], which is a Vicuna-13B [6] fine-tuned using LoRA [16]. This model achieves 92% text-layout alignment in human evaluations [7]. We treat these generated layouts as ground truths for the diffusion model.

In our method, we use both the text prompts and the layouts as guidance. Note that, our method can be applied to any layout-based diffusion model, regardless of how the layout is fused into the model. Typically, for most layout-based diffusion models, the prompts and object phrases are encoded by a text encoder, while the coordinates are encoded using Fourier embedding [28]. The features of encoded object phrases and coordinates are concatenated to construct the layout features. All encoded features are sent to the cross-attention modules in the diffusion U-Net [33].

3.1. Error Detection, Feedback Computation, and Correction

Given that the image generation with diffusion models is a dynamic denoising process, our key idea is to estimate generation errors from noisy intermediate outputs *during* the process, instead of waiting for the generation to complete. The high-level strategy is to evaluate intermediate de-noised outputs and send feedback signals to the model to correct any detected generation errors. The details of our method are shown in Figure 2. The overall process consists of 4 stages (including standard denoising):

1. **Standard Denoising:** Start with normal denoising.

2. **Error Detection:** At time step t, we use an object detector to identify all the objects in the partially denoised image at that step. The detector returns the identified objects as well as their bounding boxes.

3. **Feedback Computation:** Compare detected objects with the ground truth layouts, identify erroneous objects and categorize them in the following two buckets:

   - False positive (FP) objects: Objects whose categories or bounding boxes do not match the ground truth. These are objects that either should not be generated or are misplaced or out of proportion in size.
   - False negative (FN) objects: Objects specified in text prompt, present in ground truth layout but missing in the detection results.

   We create FP/FN layouts consisting of only FP/FN objects and their bounding boxes. The FP and FN layouts are used as one of our feedback signals. We also provide reference images as additional feedback to help the model generate missing objects (Section 3.3). Lastly, we also adjust the bounding box sizes in layouts if the scale relationship is violated (Section 3.4).

4. **Correction:** Return to a previous step k. The FP/FN layouts are separately fed to the diffusion model to generate FP/FN latent representations. The FP/FN latent representations are combined with the latent representation of the original image through composable diffusion. For FN cases, the retrieved images are fed to the model together with the FN layout (Section 3.3).

Our whole approach above is a general recipe, for which we discuss one instantiation in this paper. Each of these steps are modular and can be implemented in various ways using

different modules. Compared to approaches that fine-tune models for improving faithfulness, our approach only modifies the inference stage and therefore, significantly cheaper. Compared to most existing inference time algorithms, our approach does not involve expensive backpropagation step and is able to reduce more types of errors. Differing from methods utilized in image editing models, our approach also reduces the time it takes to correct the error, because we do not need to wait until the completion of denoising process to regenerate potentially correct images.

3.2. Composition of FP/FN Latent Representations

The idea of composable diffusion was proposed in [24]. It points out that, the sampling procedure of diffusion models (Equation (1), where ϵ is the diffusion model output) and energy-based models are functionally similar. The score-based formulation of diffusion models illustrates that learning the noise function ϵ can be interpreted as learning the score function $\nabla_x \log p_\theta(x)$, where $p_\theta(x)$ is the data distribution. EBMs learn the energy function $E_\theta(x)$ such that $\nabla_x E_\theta(x) \propto \nabla_x \log p_\theta(x)$. Hence, like energy-based models, multiple ϵ of the diffusion model can be composed to obtain a new score function, in a way shown in Equation (2).

$$x_{t-1} = x_t - \epsilon_\theta(x_t, t) + N(0, \sigma_t^2 I) \quad (1)$$

$$x_{t-1} = x_t - \sum_{i=1}^{n} \epsilon_\theta^i(x_t, t|y_i) + N(0, \sigma_t^2 I) \quad (2)$$

For example, given two input prompts, "a horse" and "woods", they can be composed as $\gamma\epsilon_\theta(x_t, t|horse) + \gamma\epsilon_\theta(x_t, t|woods) + (1-\gamma)\epsilon_\theta(x_t, t|\varnothing)$ (assuming classifier-free setting; γ is weight parameter) and the final image will have a horse in the woods. The concept can also be negated to generate an image such as the one with a horse but no woods. This method is also used by [11] and [17]. In our method, we send the FP/FN layouts to the model as two separate inputs, aside from the original layout, and get FP/FN latent representations. We compute a linear combination of them together with the latent representation of the original generation (Ori). Including the unconditioned component $\epsilon_\theta(x, t|\varnothing)$, the composed function is:

$$\hat{\epsilon_\theta}(x_t, t) = \sum_i w_i \epsilon_\theta(x_t, t|i), i \in \{Ori, FP, FN, \varnothing\} \quad (3)$$

where w_i is the guidance scale. The $\hat{\epsilon_\theta}(x_t, t)$ is on a direction that: (1) Deviates from generating FP objects (by setting w_{FP} as negative); (2) Moves close to generating FN objects (by making w_{FN} positive). This helps *remove* the FP objects or errors and *recover* the FN objects.

3.3. Image Retrieval for Missing Objects

When the model fails to generate certain objects during the initial round of generation, simply repeating the process with identical inputs is unlikely to succeed. In cases where the model supports retrieval-augmented generation, we can leverage reference images as an additional feedback to help the model generate those objects. Typically, the reference images are encoded by the CLIP vision encoder [31], and categorized based on their object types (e.g., dog, person, car). For each missing object, the corresponding visual features under its object category are retrieved. As shown in Figure 3, they are concatenated with the text and bounding box coordinate features and passed to the diffusion U-Net.

3.4. Scale Adjustment

For prompts specifying scale relationships (e.g., "the apple is larger than the clock."), we also verify the correctness of them in our intermediate detection. To determine if the prompts contain scale relationships, we simply search the prompts for key words such as "bigger", "smaller", etc. For the detected objects, we calculate their bounding box areas, compare their sizes and check if the scale relationships in the text prompt are fulfilled. If there are discrepancies, we adjust the corresponding object bounding boxes in the ground truth layouts and use it as another feedback.

For example, given objects *Obj1* and *Obj2*, and the scale relationship "*Obj1* is smaller than *Obj2*", if both are detected in the intermediate generation, we compare the size of their bounding boxes. If they do not fulfill the relationship, we switch the true bounding box of *Obj1* and *Obj2* if there is no spatial relation specified in the prompt. Otherwise, we maintain the original placements. Then, we modify the ground truth layout by shrinking the bounding box of *Obj1* by 25% and expanding the true bounding box of *Obj2* by 25%. For relationship "*Obj1* is bigger than *Obj2*", we apply similar steps but expand *Obj1* and shrink *Obj2*. If the relationship is "same", we expand the smaller bounding box to the same size as the larger one.

4. Experiments & Results

4.1. Experiment Setting

We evaluate our method using two datasets: VPEVAL [7] and HRS-Bench [1]. VPEVAL consists of programs designed to assess the quality of generations across five dimensions: object, count, spatial, scale and text rendering. We focus on the first 4 dimensions. "object" refers to the existence of an object. "count" evaluates the number of objects. "spatial" is the spatial relationship between objects. "scale" is the scale relationship between objects. The performance is measured by accuracy; the percentage of correctly generated images. HRS-Bench, similar to VPEVAL, measures a broader spectrum of skills such as count, spatial, scale, color, emotion, action, etc. We focus on count, spatial and scale dimensions in this work. Prompts of HRS are categorized into three levels of complexity: easy, medium

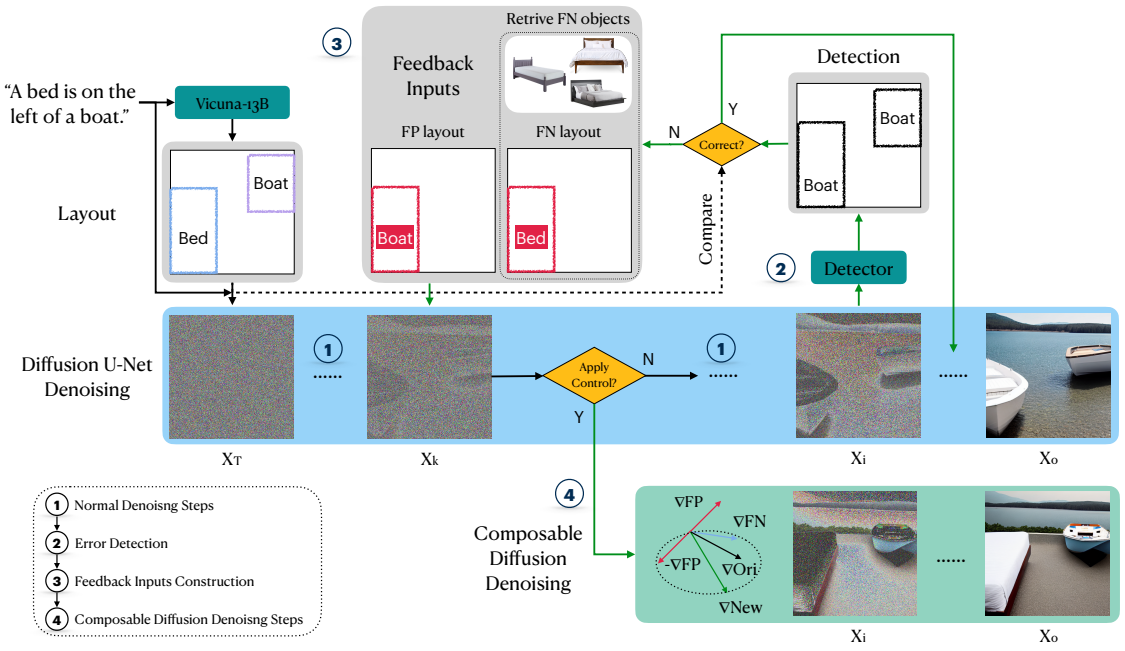


Figure 2. High level overview of our proposed approach. First, a layout generation model generates a layout for the given text prompt. Then, the prompt and the layout are both sent to the diffusion model. At denoising step i, an object detector is applied to the intermediate generation (Stage 2). The detection output is compared to the input layout and FP/FN layouts are constructed corresponding to FP/FN objects (Stage 3). The reference image features of the FN objects are retrieved and combined with the FN layout (Stage 3). We return to step k and both layouts are sent to the model as feedback inputs. Lastly, composable diffusion method is utilized to combine the latent representations of the original layout (Ori) and FP/FN layouts (Stage 4).

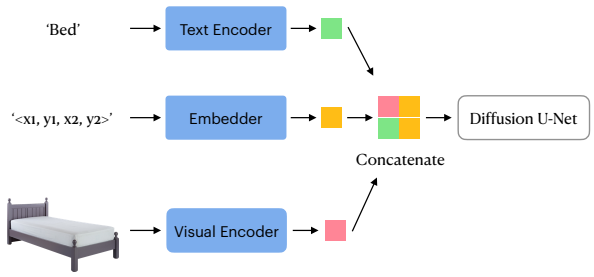


Figure 3. Fusing retrieved reference image with object label & bounding box coordinates.

and difficult levels. "Count" is evaluated using the F1 score at object level. The others are measured by the percentage of correctly generated images.

We use GLIGEN [7] as our model for layout-based generation, because GLIGEN achieves the best performance on VPEVAL. GLIGEN also supports retrieved images as input. We then apply our intervention mechanism to GLIGEN, and use YOLOv8l [18] as the error detection model. The generation process runs for 50 steps. Error detection is initiated at step 45. Upon identifying errors, the process returns to step 1. Then the feedback is applied via retrieved reference image and compositional generation until step 25. The detection-correction process is only conducted once. Following the standard practices, we employ a guidance scale of 7.5 for original layouts. For FP/FN latents, the guidance scale is adjusted to 3.75 to avoid drastic changes on the original images. Moreover, each FN object is accompanied by a single reference image. We also tune the detection timing and the duration for which the feedback is applied, exploring the balance between total runtime and accuracy.

4.2. Evaluation Results

We apply our method to the GLIGEN model and compare its performance to the unmodified version. Other models [27, 29, 30, 35] are also included in the comparison.

Table 1 shows the results on VPEVAL. With our method, GLIGEN shows notable improvements on count (+5.3%), spatial (+9%) and scale (+11.6%). Our method also outperforms other approaches by a large margin on these dimensions. Table 2 shows the results on HRS-Bench. We report the accuracy only on prompts at easy level, because as stated in the HRS paper [1], the prompts of medium and difficult level are so complex that most models have nearly zero accuracy on spatial and scale subsets. In Table 2, our method improves GLIGEN by 10.2% on count, 4.5% on spatial and 1.7% on scale. Our method outperforms other

Model	Object	Count	Spatial	Scale	Avg
GLIGEN	97.5	74.7	53.3	28.9	63.6
Dalle-Mega	94.0	45.6	17.0	8.5	41.3
SDXL	97.5	51.3	42.8	22.6	53.6
[35]	97.0	74.5	58.3	29.9	64.9
[27]	97.3	78.5	57.1	29.6	65.6
[30]	98.5	66.6	53.4	30.7	62.3
Ours	**98.3**	**80.0**	**62.3**	**40.5**	**70.3**

Table 1. Model comparisons on VPEval.

Model	Count	Spatial	Scale
GLIGEN	73.4	54.2	61.1
DALL.E 2	81.9	28.3	29.9
[35]	81.7	55.7	55.7
[29]	71.8	54.2	39.7
[27]	83.8	**60.8**	56.3
Ours	**84.6**	58.7	**62.8**

Table 2. Model comparisons on HRS.

Model	Time (s)
GLIGEN	7.15
[35]	18.69
[27]	22.55
Ours(fastest/reported)	12.62/16.63

Table 3. Running time comparison.

works by a large gap on both datasets. In Figure 4 and 5, we show examples generated by GLIGEN with and without our intervention-based generation approach. These figures illustrate the errors that the original model exhibits but get fixed by our approach. Lastly, in Appendix, we also include a detailed analysis, show more results/examples and compare our method with error correction via image editing.

4.3. Accuracy-Latency Tradeoff

A key consideration on our evaluation is balancing runtime latency with faithfulness of generation. The incorporation of a detection model and reverting to an earlier step enhances faithfulness but at the expense of increased overall running time of the algorithm. In Table 3, we compare the running times of GLIGEN alone and GLIGEN with our method for the fastest setting and the setting in Table 1 and Table 2. Despite our method takes longer time, it remains more efficient than other methods.

There are two critical hyperparameters: detection timing and the control duration. Early detection on noisier images can incur more errors, whereas later detection, though more accurate, extends the overall running time. Longer control duration can lead to more faithful generations, albeit at the expense of increased running time. Figure 6 shows the VPEVAL performance (measured by average accuracy

Detectors	Object	Count	Spatial	Scale
GroundingDINO	97.00	80.30	57.20	37.90
YOLOV8l	97.75	81.10	59.10	36.90

Table 4. Ablation study on different detectors, evaluated on VPEVAL. Retrival is not applied.

Detectors	Object	Count	Spatial	Scale
No Retrival	97.8	81.1	59.1	36.9
Retrival	98.3	80.0	62.3	40.5

Table 5. Ablation study on image retrival, evaluated on VPEVAL.

of object, count, spatial and scale) under different detection time points. For $t = 15$ and $t = 25$, the results are close, which is probably because the detection accuracy is low at early stage. However, accuracy improves when the detection is postponed to $t = 35$ and $t = 45$. Delaying the detection by every 10 steps results in an increase in the mean running time of approximately 1.2s-1.5s.

Figure 7 shows the average VPEVAL accuracy under different control durations. When the length increasese from 15 to 25, the average accuracy increases by around 1%. A notable observation is that there is no significant improvement in accuracy when the control duration is extended further. This can be attributed to the fact that by the later stages, the key features of objects have already been rendered, diminishing the necessity for additional controls. Furthermore, when the control duration increases by 10 steps, the mean running time rises by around 1.4s-1.8s.

Lastly, note that in Figure 6 and Figure 7, even we use the fastest settings, the average accuracy 66.3 is still higher than the baseline GLIGEN (63.6) and other models.

4.4. Ablation Studies

We run ablation on detectors. We change the detector to GroundingDINO [25] and run our method on VPEVAL. The results are shown in Table 4. YOLOV8l performs slightly better than GroundingDINO on average.

We also run ablation on image retrival. Table 5 shows the VPEVAL results with/without retrival. Compared with no retrival, using image retrival shows improvement on object, spatial and scale. In Appendix, we also show images generated with and without retrival.

5. Conclusion and Discussion

In conclusion, we propose a strategy to control the denoising process of diffusion models through monitoring and intervention. The generation errors are early detected and corrected during the generation process itself. Results on VPEVAL and HRS demonstrate that our approach provides significant improvement.

One concern of our work might be the sacrifice of run-

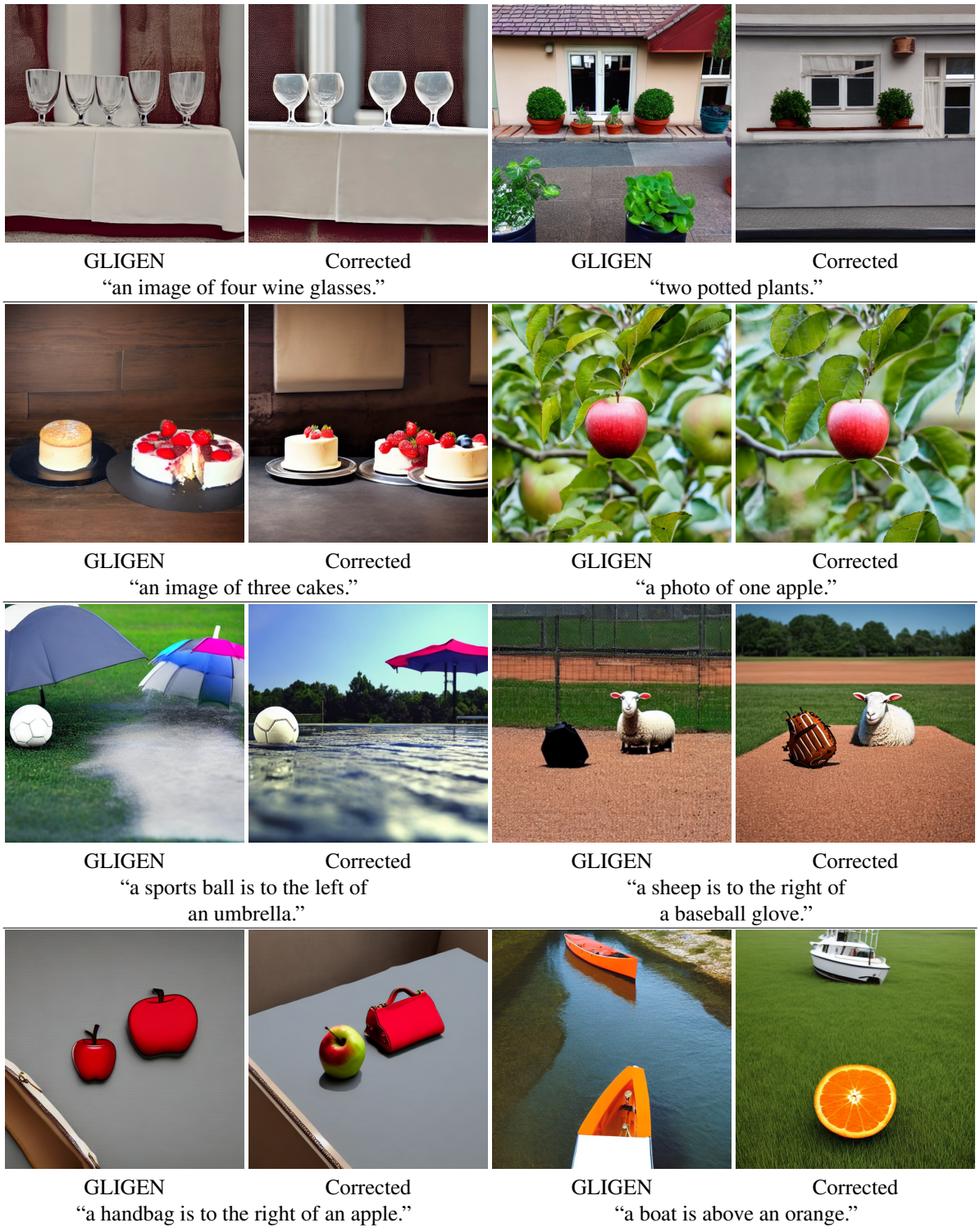


Figure 4. Generation examples after applying intervention and composition.

ning time. Although we use around 9s in exchange for +6.7% average accuracy, our method is still faster than other existing works. On the other hand, in real-world applications, one may apply our method only to cases whose initial generations contain errors. Our time-accuracy tradeoff is acceptable for error correction. For future research, the time could be further reduced by improving the detection accuracy on noisy images and tuning the guidance scales.

Figure 5. Generation examples after applying intervention and composition.

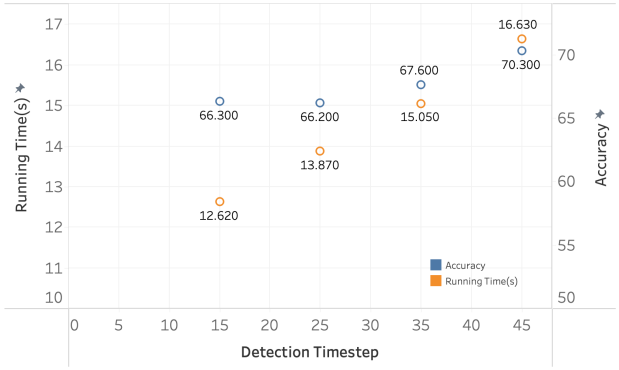


Figure 6. Average VPEVAL accuracy comparison for detecting at different time step (total t=50) and controlling from t=0 to t=25.

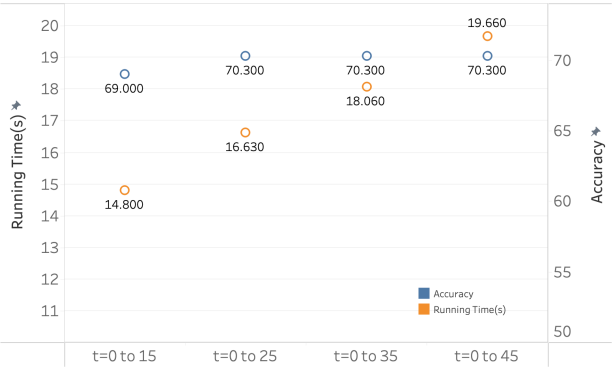


Figure 7. Average VPEVAL accuracy comparison for applying control for various length of steps. Detection is at t=45.

To compose the FP/FN generation with the original generation, we use linear combination with preset weights. [9] pointed out that, the linear combination is a mathematical approximation of the true score of the diffusion target and sometimes it leads to subpar generations. At each denoising step, it uses Markov Chain Monte Carlo to get the final score function. However, we found that it is not applicable to high-resolution images because it requires extensive backpropagation steps, which is expensive on large images. Developing a more accurate and time-efficient way of composition would be a direction of future research.

In our work, we operate under the assumption that the generated layout is correct in the majority of cases. However, the accuracy peaks around 90% on average [7], setting a performance ceiling for our strategy. Besides improving the layout generation model, a more promising direction would be enabling the model to generate correct images despite incorrect layouts. The diffusion model should learn if the input layout can be trusted or not. If not, there should be an alternative method to generate the correct image.

References

[1] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models, 2023. 4, 5

[2] Betker et al. Improving image generation with better captions, 2023. 3

[3] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 2023. 2

[4] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *WACV*, 2024. 2

[5] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. Re-imagen: Retrieval-augmented text-to-image generator. In *ICLR*, 2023. 2

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 3

[7] Jaemin Cho, Abhay Zala, and Mohit Bansal. Visual programming for text-to-image generation and evaluation. In *NeurIPS*, 2023. 2, 3, 4, 5, 8

[8] Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *ICCV*, 2023. 2

[9] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *ICML*, 2023. 8

[10] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv:2306.00986*, 2023. 2

[11] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv:2212.05032*, 2023. 4

[12] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, Li Chen, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *CVPR*, 2023. 2

[13] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022. 2

[14] Biao Gong, Siteng Huang, Yutong Feng, Shiwei Zhang, Yuyuan Li, and Yu Liu. Check, locate, rectify: A training-free layout calibration system for text-to-image generation. *arXiv:2311.15773*, 2023. 3

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[16] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 3

[17] Vikram Jamwal et al. Composite diffusion: whole¿= sparts. In *WACV*, 2024. 4

[18] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 2023. 5

[19] Wonjun Kang, Kevin Galim, and Hyung Il Koo. Counting guidance for high fidelity text-to-image synthesis. *arXiv:2306.17567*, 2023. 2

[20] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. If at first you don't succeed, try, try again: Faithful diffusion-based text-to-image generation by selection. *arXiv:2305.13308*, 2023. 3

[21] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv:2302.12192*, 2023. 2

[22] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. *CVPR*, 2023. 2

[23] Luping Liu, Zijian Zhang, Yi Ren, Rongjie Huang, Xiang Yin, and Zhou Zhao. Detector guidance for multi-object text-to-image generation. *arXiv:2306.02236*, 2023. 2, 3

[24] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *ECCV*, 2022. 4

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv:2303.05499*, 2023. 6

[26] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *ICCV*, 2023. 3

[27] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023. 3, 5, 6, 1

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 2021. 3

[29] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. *arXiv:2306.05427*, 2023. 5, 6, 1

[30] Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation. In *Proceedings of*

the 31st ACM International Conference on Multimedia, MM '23, page 643–654, New York, NY, USA, 2023. Association for Computing Machinery. 2, 5, 6, 1

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[32] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 1

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 1

[35] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *ICCV*, 2023. 2, 5, 6, 1

[36] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Reco: Region-controlled text-to-image generation. In *CVPR*, 2023. 2

[37] Zhiyuan Zhang, Zhitong Huang, and Jing Liao. Continuous layout editing of single images with diffusion models. *arXiv:2306.13078*, 2023. 2

[38] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, 2023. 2

A. Appendix

A.1. Experiment Details and Result Analysis

In Table 1 and Table 2, we compare our model with other models/methods. The comparison between GLIGEN and GLIGEN with our methods shows that our methods help improve the performance of layout-based generation models. The performance of DALL.E and SDXL is lower because these models are not conditioned on layouts. Their low performance shows that, the faithfulness of text-to-image generation is still a problem and the layout could improve the faithfulness. [35], [30] and [29] are three methods that improve the faithfulness of layout-based image generation models. [35] and [29] first compute the gradients with respect to a pre-defined loss function during the generation process, and then move the latents towards the gradients. [30] improves the transformer encoder to better encode the layout information. [27] focuses on image editing. Nevertheless, its methods can be applied to improve the faithfulness of generation - through detecting the generation errors and manipulating the initial noise to correct those errors. In our experiments, [35], [29] and [27] are applied to GLIGEN, same as our method. [30] is also a modified GLIGEN.

On VPEVAL, our method acquires the most gain on Scale and Spatial. The main reason is that, both object errors and relationship errors are counted in scale/spatial metrics. For example, for the prompt "a sheep to the right of a baseball glove", missing the baseball glove will also be counted as an error under spatial metric in the VPEVAL evaluation pipeline. Our methods reduce both object errors and relationship errors. Hence, the improvement is larger. For VPEVAL object, the accuracy of all models is high because the prompts are easy, which only ask the model to generate one object. On HRS, the improvement on count is high while that on spatial and scale is lower. We think it is because of the difficulty level of its prompts.

A.2. Inception Scores

We computed the Inception Scores for the original GLIGEN and the GLIGEN with our strategies, using the generated images on VPEval. The results are in Table 6. After applying our methods to GLIGEN, the score only changes slightly, indicating that the quality trade-off is minimal. We did not compute FID because we do not have reference real images for VPEval and HRS.

	IS
GLIGEN	26.21±3.31
GLIGEN with our methods	26.35± 4.03

Table 6. IS comparison

A.3. Additional Images

Figure 8 and 9 show generations with standard GLIGEN model, GLIGEN combined with our intervention-based inference method but without retrieval-based feedback, and GLIGEN with our full approach. Comparing the images in second and third column, one can find that, sometimes the model fails to generate the target object only with the text features. The use of retrieved image features helps the model to generate the missing objects.

A.4. Error Correction through Image Editing

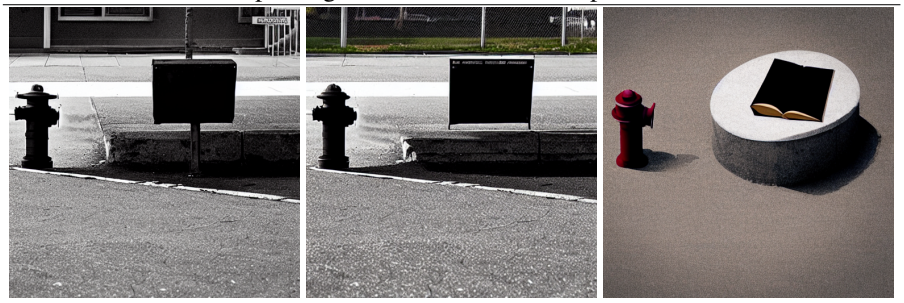
We also run image-editing model InstructPix2Pix repeatedly to correct the generation errors. The image guidance scale is set as 1.5. The model runs for 4 times and the average running time of each is around 5s (20s in total). At each turn, the model tries to correct the image generated in the previous turn. The editing prompt is designed by human, tailor-made to correct the errors of the input image. In Figure 10, we show the editing results for selected cases in Figure 4 and 5. From the figures one can see that the model fails to correct most errors in 4 turns. These examples show several disadvantages of "correcting by image editing", compared with our methods.

- The image editing model usually needs to run several times to correct the errors, which largely increases the running time.
- The same guidance scale cannot be applied to all images. For some cases, our guidance scale (1.5) leads to trivial changes.
- The model often fails to understand instructions about locations or scales.

Note that these problems are for InstructPix2Pix. In the future there may be an improved image editing model that addresses these problems.

GLIGEN No Retrieval Retrieval

"a parking meter is to the left of a person."



GLIGEN No Retrieval Retrieval

"a book is to the right of a fire hydrant."

GLIGEN No Retrieval Retrieval

"a clock and an apple. the clock is smaller than the apple."

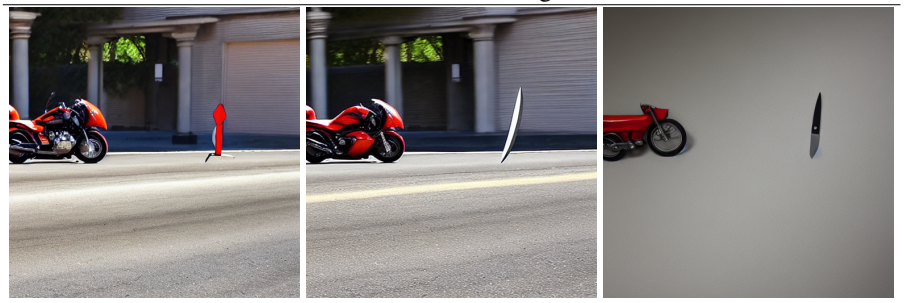
GLIGEN No Retrieval Retrieval

"a bicycle and a toothbrush. the bicycle is bigger than the toothbrush."

Figure 8. Comparing image generations with GLIGEN, GLIGEN with our proposed approach but without retrieval-based feedback, and GLIGEN with our full approach.

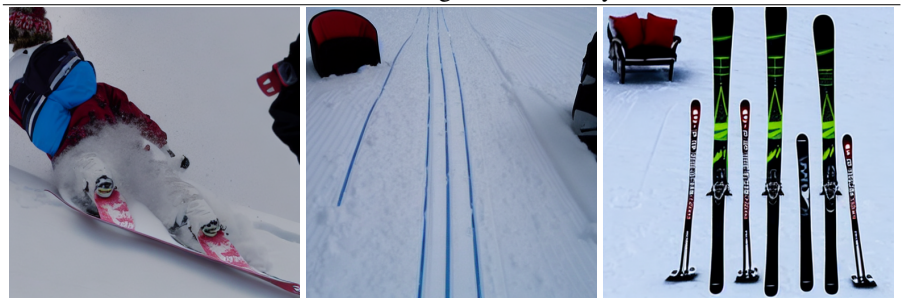
GLIGEN No Retrieval Retrieval

"a bus is below a wine glass."



GLIGEN No Retrieval Retrieval

"a knife is to the right of a motorcycle."



GLIGEN No Retrieval Retrieval

"a couch and a skis. the couch is smaller than the skis."

GLIGEN No Retrieval Retrieval

"a cup and a motorcycle. the cup is bigger than the motorcycle."

Figure 9. Comparing image generations with GLIGEN, GLIGEN with our proposed approach but without retrieval-based feedback, and GLIGEN with our full approach.

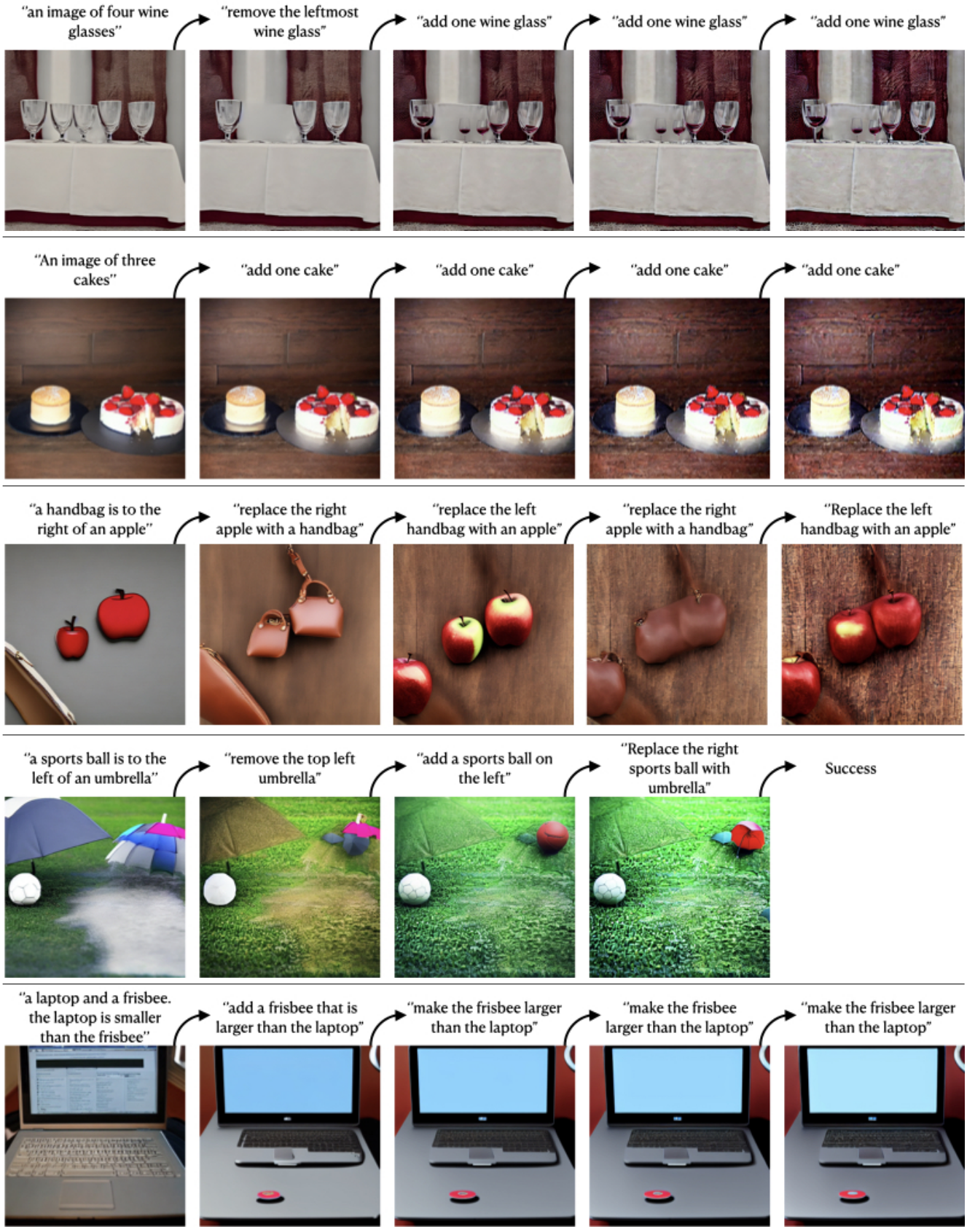


Figure 10. Correcting generation errors via image editing with InstructPix2Pix. In each example, we run the model for 4 times. For each run, we manually design the editing prompt based on the output of the previous run. The editing prompt is shown above the images and used to edit the image at previous stage. The corresponding output is shown below the prompt.