VODiff: Controlling Object Visibility Order in Text-to-Image Generation

Supplementary Material

A. Overview

In this supplementary material, we provide additional details, experimental analyses and extension applications to support and expand upon the findings presented in the main paper.

Specifically, We address several key aspects in our supplementary materials: discussing similarities and key differences with concurrent works and customized T2I models (Sec. B); Offering additional clarifications on our approach, including experimental configurations and supplementary ablation analyses (Sec. C–Sec. D); Presenting additional experiments to validate our method, including further comparisons with state-of-the-art approaches (Sec. E); Demonstrating extensions and applications of our method by integrating it into different pre-trained models and exploring new applications (Sec. F).

B. Discussion with Related Works

In sampling-based methods, while concurrent work, LRDiff [48] similarly enhances the generation of objects in specific regions by injecting noise into targeted areas to provide cues for the score estimation network, thereby guiding the denoising process toward generating a single visual concept within a specified region. However, although they apply distinct guidance to each object, the visibility order between objects is not taken into account.

To address this issue, our method adopts a fundamentally different approach: SDP synthesizes objects sequentially, stage by stage, following the visual order from bottom to top. It dynamically adjusts the guidance scale, allowing the denoising network to focus on different objects at each stage and ensuring the establishment of correct occlusion relationships. Additionally, we propose a visibility order-aware loss that optimizes the cross-attention maps to address the issue of local shifts that may arise when using SDP alone, thereby enhancing the accuracy of object positioning and occlusion relationships in the generated images.

Customized T2I models Unlike layout-guided T2I methods, customized T2I models aim to integrate specific input objects into the generated images while preserving their identity features. To support customized generation, they accept specific object inputs to customize the identity of objects, as well as provide attribute and spatial control through text prompts and masks.

Optimization-based methods [6, 14, 24, 29, 44, 55, 74] can achieve high-fidelity identity preservation. However, they are slow and may suffer from overfitting occasionally. In contrast, encoder-based methods [9, 28, 32, 45, 60, 61, 65, 71, 76, 77, 79] enable zero-shot performance but may either lose the identity of the object or produce trivial results resembling copy-pasting.

When integrating input objects into specific regions of an image, these methods focus on ensuring that the generated objects match the identity of the inputs and the attributes defined by the prompts. However, they do not consider the visibility order between the integrated object and other objects within or near that region. This limitation hinders their effectiveness in dealing with occlusion issues.

C. Experimental Settings

C.1. Implementation Details

We utilize Stable Diffusion 1.5 [54], trained on the LAION dataset [57]. The coefficients for the VOA loss are set as $\alpha_o = 6$, $\alpha_{vis} = 4$, and $\alpha_{bac} = 2$, with the optimization coefficient α defaulted to 0.1. The initial guidance scale a is set to 1.

All experiments are conducted on a single NVIDIA RTX 4090 GPU. Unless otherwise stated, we adopt the DDIM sampler [58] with 50 sampling steps for the reverse diffusion process, using a fixed classifier-free guidance scale of 7.5.

C.2. Preprocess of the Attention Maps

To compute the loss between the attention map $\mathbf{A}_{t,n}$ and each specific region, we first extract the cross-attention maps associated with the object prompt c_n from each layer. After discarding the attention maps linked to the <sot> token, we apply a Softmax operation to the remaining maps. Subsequently, Gaussian smoothing is applied, following the approach in [5], to produce $\mathbf{A}_{t,n}$ at a resolution of 16×16.

C.3. VOBench

As illustrated in Fig. 1, our VOBench comprises three components: bounding boxes with visibility order, reference images, and text prompts. It includes 200 combinations constructed through the following process. All raw images in VOBench are meticulously curated from the Internet. Initially, we employ GroundingDINO [34] and GPT-4V to generate bounding boxes and corresponding text prompts for each object in the images. Subsequently, these bounding boxes and prompts are manually verified, and a visibility order is assigned to each object. The benchmark is further categorized into four groups, containing 2–5 objects per image. For each category, 50 images are manually selected to ensure diversity in occlusion relationships, prioritizing variations in the degree of overlap between bounding boxes.

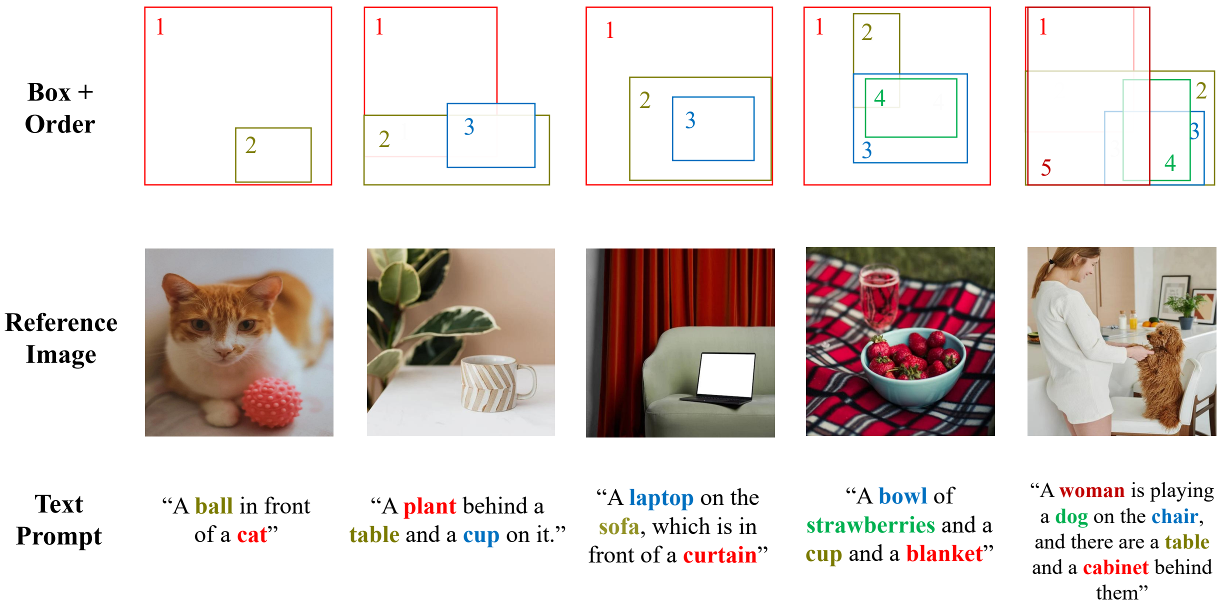

Box + Order
1
2
1
2
3
1
2
3
1
2
4
3
1
2
3
5
4
Reference Image
Text Prompt
"A ball in front of a cat"
"A plant behind a table and a cup on it."
"A laptop on the sofa, which is in front of a curtain"
"A bowl of strawberries and a cup and a blanket"
"A woman is playing a dog on the chair, and there are a table and a cabinet behind them"

Figure 1. Visual illustration of our VOBench.

D. Additional Ablation Studies

D.1. Smooth Guidance for Sequential Denoising Process

Our SDP employs a smooth guidance mechanism, in addition to assigning strong guidance to the objects at the current stage, we also apply low-intensity guidance to objects not belonging to that stage (*i.e.*, objects will be mainly handled at other stages). This approach is based on the observation that object outlines are typically formed during the early steps of the diffusion process, particularly before $T = 20$ [8].

As shown in Fig. 2, we visualize the results of using smooth guidance compared to applying guidance exclusively to a single object at each stage. The second row illustrates the results when visual guidance is applied to only one object at a time, while the first row shows the results when smooth guidance is applied across different objects. It can be observed that focusing exclusively on a single object n at each stage may result in failures to generate other objects. In contrast, the smooth guidance effectively addresses this issue, enabling the successful synthesis of all objects.

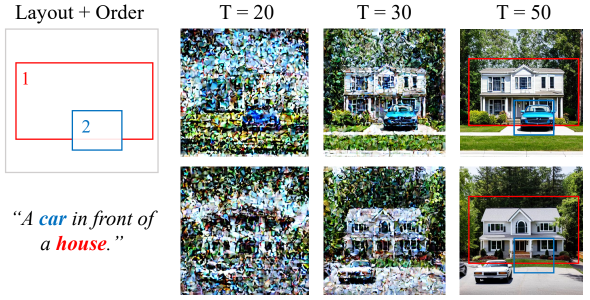


Figure 2. Focusing exclusively on a single object n at each denoising stage can lead to the failure of generating other objects. Row 1 illustrates results with smooth guidance, while row 2 shows those without it.

D.2. Influence of Local Shifts

In the main paper, we discussed that while SDP can constrain target generation within a specified region through visual guidance, consecutive convolution layers in the denoising model may introduce local shifts, potentially causing changes in object positions and inaccurate occlusion relationships.

For instance, as shown in the first row of Fig. 4, the bear shifts to the right within the image. Although the occlusion relationships remain unchanged, the spatial positions of the objects are adjusted. Additionally, when the visual guidance for object n shifts away from object $n + 1$, as illustrated in the second row (where the apple moves further from the bag), both the spatial positions and occlusion relationships may be altered.

To address this problem, we propose constraining spatial positioning and occlusion relationships at the feature level by introducing a visibility order-aware loss, which optimizes the cross-attention maps. As demonstrated in the

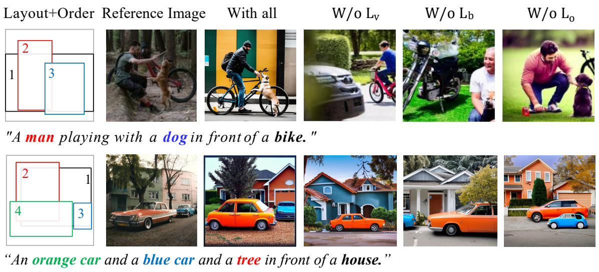

Layout+Order
Reference Image
With all
W/o L_v
W/o L_b
W/o L_o
1
2
3
"A man playing with a dog in front of a bike."
1
2
3
4
"An orange car and a blue car and a tree in front of a house."

Figure 3. Visual results of ablation study on VOA loss

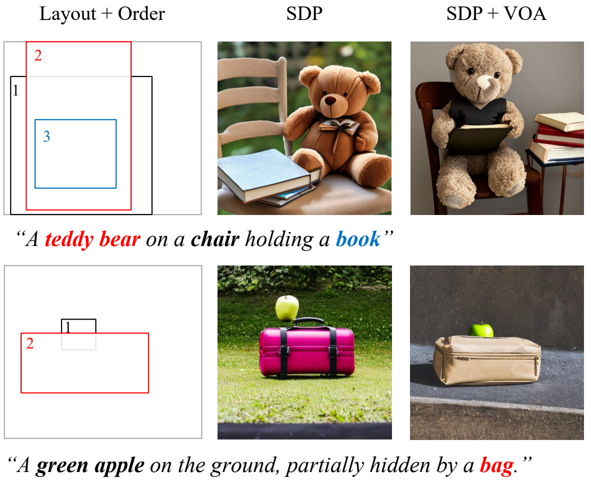

Layout + Order
SDP
SDP + VOA
1
2
3
"A teddy bear on a chair holding a book"
1
2
"A green apple on the ground, partially hidden by a bag."

Figure 4. Illustration of local shifts introduced by SDP constraints.

second and third columns of Fig. 4, our VOA loss effectively addresses the issue of local shifts, significantly improving the accuracy of occlusion relationship generation.

D.3. Denoising Steps

Since our SDP divides the entire denoising process into stages, increasing the number of objects reduces the steps per stage, potentially affecting the quality of the generated images. To address this, we evaluated the impact of increasing sampling steps using a curated benchmark of 150 images (5–7 objects per image, with 50 images per category) and calculated results for four metrics to determine the optimal number of steps for generating images with a higher object count. As shown in Table 1 and Fig. 5, when T increases to 75 steps, although FID shows a slight decline, the LA metric improves from 40.08 to 43.25, reflecting more precise layout constraints during object generation. Notably, OA steadily increases from 27.00 to 39.50, as illustrated by the comparison between the second and third columns in Fig. 5, indicating enhanced handling of occlusion relationships. Similarly, the Clip-Score shows a slight improvement, suggesting better alignment with the textual prompt. However, when $T > 75$, all metrics decline, resulting in suboptimal performance, as shown by the comparison between the third and fourth columns in Fig. 5.

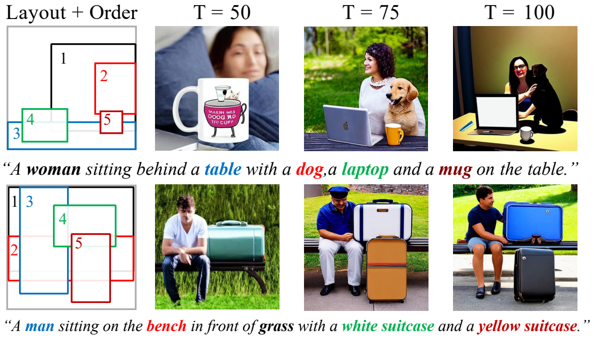

Layout + Order
T = 50
T = 75
T = 100
1
2
3
4
5
"A woman sitting behind a table with a dog,a laptop and a mug on the table."
1
2
3
4
5
"A man sitting on the bench in front of grass with a white suitcase and a yellow suitcase."

Figure 5. Visual results of ablation study on time step T.

However, as the number of objects continues to increase, our method begins to face challenges. We hypothesize that a primary reason lies in the increased complexity of the optimization process, which necessitates the simultaneous optimization of a greater number of attention maps.

Table 1. Test on 4 metrics of increasing denoising steps T when generating images with high number of objects.

T	FID ↓	Clip-Score ↑	LA ↑	OA ↑
50	23.35	28.91	40.08	27.00
60	**22.52**	29.03	42.91	32.50
70	23.18	29.17	42.97	38.50
75	23.26	**29.19**	**43.25**	**39.50**
80	23.41	29.05	43.17	39.00
90	24.68	28.92	39.84	37.50
100	26.34	28.67	38.17	34.50

D.4. Guidance Scale

In SDP, the choice of the guidance scale is a crucial factor influencing the quality of generated results. To optimize our method's performance, as shown in Table 2 and Fig. 6, we tested different initial values on our VOBench and found that selecting $a/2$ as the initial value yields the best performance across various metrics and image generation quality. If the guidance scale value is too low (e.g., $a_0 = 0$ in column 2), the sampling process during each object's phase focuses more on the object itself, neglecting other objects. Conversely, if the guidance scale is too high (e.g., $a_0 = 0.8a$ in column 7), it interferes with the sampling direction of the current object, reducing the accuracy of the object's generated position.

We compared the quality of the results generated for different numbers of objects using various guidance scale values. It can be observed from Table 3 that as the number of objects increases, the initial guidance value has a greater

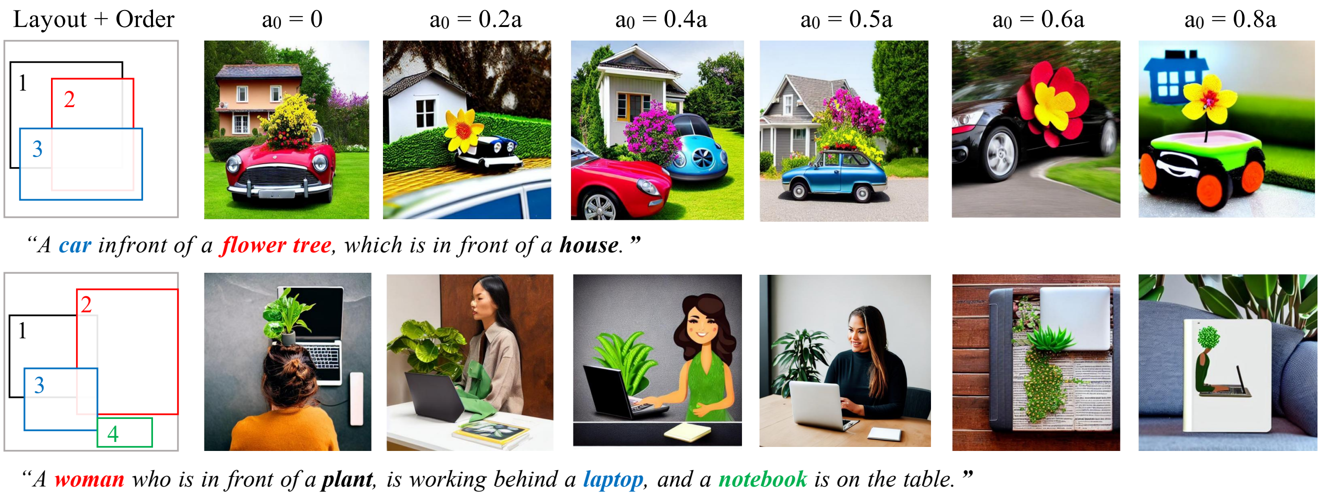

Layout + Order
a0 = 0
a0 = 0.2a
a0 = 0.4a
a0 = 0.5a
a0 = 0.6a
a0 = 0.8a
1
2
3
"A car infront of a flower tree, which is in front of a house."
1
2
3
4
"A woman who is in front of a plant, is working behind a laptop, and a notebook is on the table."

Figure 6. Evaluation on different initial guidance scale.

Table 2. Quantitative comparison of different initial guidance scale values

Guidance Scale	FID↓	CLIP-Score↑	LA↑	OA↑
$a_0 = 0$	16.89	29.66	42.28	74.50
$a_0 = 0.2a$	10.51	29.65	42.35	76.50
$a_0 = 0.4a$	10.29	29.68	44.13	73.00
$a_0 = 0.5a$	**10.03**	**29.73**	**55.11**	**82.50**
$a_0 = 0.6a$	11.09	29.70	45.35	76.50
$a_0 = 0.8a$	13.39	29.71	43.35	73.50

Table 3. Quantitative comparison of different initial guidance scale values for different numbers of objects

Guidance Scale	OA (2 objects)↑	OA (3 objects)↑	OA (4 objects)↑	OA (5 objects)↑
$a_0 = 0$	90.00	74.00	68.00	66.00
$a_0 = 0.2a$	88.00	82.00	72.00	68.00
$a_0 = 0.4a$	**90.50**	86.00	74.00	68.50
$a_0 = 0.5a$	90.00	**88.00**	**78.00**	**74.00**
$a_0 = 0.6a$	88.00	82.00	70.00	66.00
$a_0 = 0.8a$	86.00	78.00	68.00	64.00

impact on the correctness of the visual order of the generated objects. When the guidance scale is set to $a/2$, it performs better than other settings for 3–5 objects. Thus, we set the guidance scale to $a/2$ by default.

E. Comparison to SOTAs

E.1. Inference Time

The core of our SDP process is to divide the original sampling steps into distinct stages for different objects, allowing the application of tailored noise guidance. This approach enables us to achieve superior results in object occlusion compared to existing methods, without increasing the sampling steps per object. As shown in Table 4, we evaluated the inference time of various training-free methods by calculating the average inference time per image on VOBench. Our method is competitive in inference time, with a sampling speed second only to DenseDiffusion[27]. This is because DenseDiffusion does not perform iterative optimization of z_t based on the attention map. Nevertheless, our method significantly outperforms others in handling occlusion relationships between objects.

E.2. Comparisons on COCO-based benchmark

To further demonstrate the effectiveness of our method, we constructed a COCO-based benchmark by selecting a subset of images from the COCO dataset. The creation process is as follows: we first analyze the number of entities in the text prompts from the annotations of the COCO dataset. Next, we employ GroundingDINO [34] to detect the bounding box corresponding to each entity. To ensure that each image contains complex occlusion relationships, we calculate the number of overlapping bounding boxes and the size of their overlapping areas. Finally, we randomly select 25 images from each category, classified by the number of objects (ranging from 2 to 5), with varying degrees and sizes of occlusions, resulting in a benchmark comprising 100 images.

We conduct extensive comparisons on this dataset and present the results in Table 5. Our method demonstrates superior performance on the COCO-based benchmark, particularly in handling overlapping objects. While our FID score of 8.43 is slightly higher than the training-based method MIGC (8.23), we achieve the highest scores in both location accuracy (LA) and occlusion accuracy (OA), with LA at 65.83 and OA at 86.50. This indicates that our method

Table 4. Comparison of the inference time of our method with other training-free methods on one RTX4090 GPU.

Method	BoxDiff[69]	R&B[67]	AR.[46]	MultiDiff.[3]	DenseDiff.[27]	RPG-Diffusion[72]	FreeControl[38]	Ours
Inference Time (s)	37.09	38.52	52.91	35.44	**28.72**	48.39	47.02	35.41

Table 5. Evaluation results of various methods on the COCO-based benchmark. The best and second-best performances are marked in **bold** and underlined, respectively.

Metrics	Training-based					Training-free							
	SmartCtrl [35]	ControlNet [78]	MIGC [82]	InstanceDiffusion [64]	AnyControl [62]	BoxDiff [69]	R&B [67]	AR [46]	MultiDiffusion [3]	DenseDiffusion [27]	FreeCtrl [38]	RPG-Diffusion [72]	**Ours**
FID ↓	10.33	9.72	**8.23**	8.51	9.01	21.99	14.17	17.59	18.42	16.68	15.19	10.12	8.43
CLIP-Score ↑	29.72	29.82	29.89	29.93	29.87	29.51	28.66	29.21	28.51	29.45	29.44	29.51	**29.95**
LA ↑	62.85	59.73	65.62	63.58	64.37	16.21	27.88	23.57	22.72	32.49	36.39	48.37	**65.83**
OA ↑	60.50	61.50	75.50	75.00	77.50	22.50	27.50	31.50	26.50	23.50	26.50	54.50	**86.50**
AR-Q ↓	7.17	5.88	5.06	3.93	3.91	10.18	8.81	8.47	9.04	12.19	10.97	3.44	**1.96**
AR-L ↓	3.98	5.07	2.30	3.61	3.76	12.18	11.40	10.01	10.58	9.89	9.75	6.81	**1.67**
AR-O ↓	4.44	5.49	3.88	3.28	2.72	12.84	9.83	12.18	7.63	10.62	10.00	6.55	**1.54**

excels in accurately positioning objects and modeling their occlusion relationships, even without training on large-scale datasets. Furthermore, our method outperforms all others in the User Study metrics, achieving the best average rankings in quality (AR-Q: 1.96), layout accuracy (AR-L: 1.67), and occlusion handling (AR-O: 1.54). These results underscore the effectiveness of our approach in generating high-quality images that adhere to provided layouts and accurately represent complex occlusions among objects.

E.3. Visual Comparisons with Other Methods

In the main paper, we compared our method with six state-of-the-art (SOTA) methods. Here, we extend the comparison to include six additional SOTA methods. As shown in Fig. 7, our method demonstrates superior spatial consistency and occlusion handling. For instance, in the second row, methods (b–f) fail to correctly handle the visibility order of objects, while in the third row, methods (b–g) exhibit positional shifts of objects.

In addition, in scenarios where different visibility orders apply to different parts of an object's area, our method can also handle these cases well. We compared our approach with existing methods on two representative cases, as shown in Fig. 8. In the first case, depicted in the top row, our method successfully avoids missing objects and generates an image with the correct occlusion relationship between the woman and the phone, outperforming other methods [2]. In the second case, shown in the bottom row, some methods fail to preserve the connection between the umbrella handle and the hand (columns d and e). In contrast, our method accurately captures the occlusion relationships between different parts of the person and the umbrella, ensuring correct positioning and occlusion across all objects.

F. More Results of Our *VODiff*

To further validate the effectiveness of our method, as shown in Fig. 9, we first present more results generated using our approach on SD1.5 [54]. These results demonstrate that our method achieves high-quality image generation, precise spatial control, and effectively handles occlusion relationships between objects.

Furthermore, to illustrate the generalizability of our method, we integrate it into SDXL [47], an upgraded version of Stable Diffusion known for producing higher-quality images and offering a more accurate understanding of prompts. Visual results in Fig. 10 show that leveraging the enhanced pretrained model significantly improves the quality of the generated images, as well as the accuracy of spatial positioning and occlusion relationships.

F.1. Enhancement of Existing Methods Using Our *VODiff*

Since our *VODiff* require no training and can be seamlessly integrated into many existing diffusion-based models [30, 64, 78, 82], we conducted experiments to demonstrate that our approach significantly improves the accuracy of occlusion relationship generation in current pretrained models. As shown in Table 6, we compare the performance of four different methods (GLIGEN [30], ControlNet [78], MIGC [82], and InstanceDiffusion [64]) on VOBench, both with and without the integration of *VODiff*. Across all models, integrating *VODiff* leads to substantial improvements in FID scores (indicating better image quality) and enhances the accuracy of positional and occlusion relationships, as reflected by increases in LA and OA. For instance, in the case of MIGC, the FID score improves from 9.82 to 7.74, while LA and OA rise from 54.62 to 63.75 and from 65.00 to 90.50, respectively.

As shown in Fig. 11, Each case is represented by two rows: the results before and after applying our method, respectively. Our *VODiff* help these training-based methods address issues such as object disappearance and incorrect

[2] Note that all methods compared here adopted the Stable Diffusion 1.5 as the foundation model.

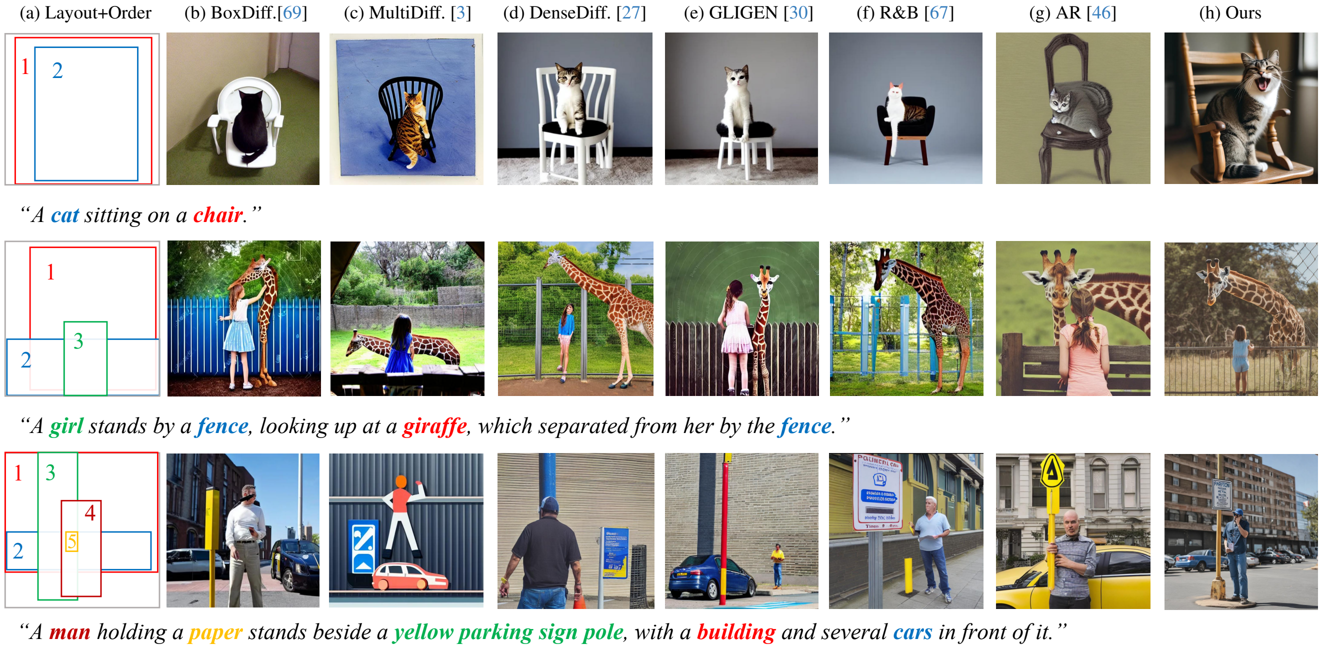

(a) Layout+Order
(b) BoxDiff.[69]
(c) MultiDiff. [3]
(d) DenseDiff. [27]
(e) GLIGEN [30]
(f) R&B [67]
(g) AR [46]
(h) Ours
1
2
"A **cat** sitting on a **chair**."
1
2
3
"A **girl** stands by a **fence**, looking up at a **giraffe**, which separated from her by the **fence**."
1
3
4
2
5
"A **man** holding a **paper** stands beside a **yellow parking sign pole**, with a **building** and several **cars** in front of it."

Figure 7. Comparison with other six SOTA methods

positional and occlusion relationships. This demonstrates that incorporating *VODiff* consistently enhances both the quality of the generated images and the precision of their spatial and occlusion relationships.

F.2. More Applications of Our Method

Object insertion. Our SDP method can be combined with Inversion [58] to enable object insertion, ensuring that the inserted objects adhere to the preset visibility order relative to the original objects in the image, while preserving the background unchanged. First, we extract the masks of the objects of the original image. Then, following the region separation approach described in the main paper, we com-

Table 6. Our *VODiff* can be integrated into various pre-trained models, enhancing the performance of existing methods across four metrics.

Method	SDP&VOA	FID↓	CLIP-Score↑	LA↑	OA↑
GLIGEN[30]	×	11.27	29.09	54.85	55.50
	✓	8.81	29.91	57.52	86.00
ControlNet[78]	×	18.91	29.34	52.73	51.50
	✓	9.03	29.97	60.21	86.50
MIGC[82]	×	9.82	29.61	54.62	65.00
	✓	7.74	30.12	63.75	90.50
InstanceDiff.[64]	×	10.21	29.53	53.58	63.50
	✓	9.61	29.98	62.44	88.50

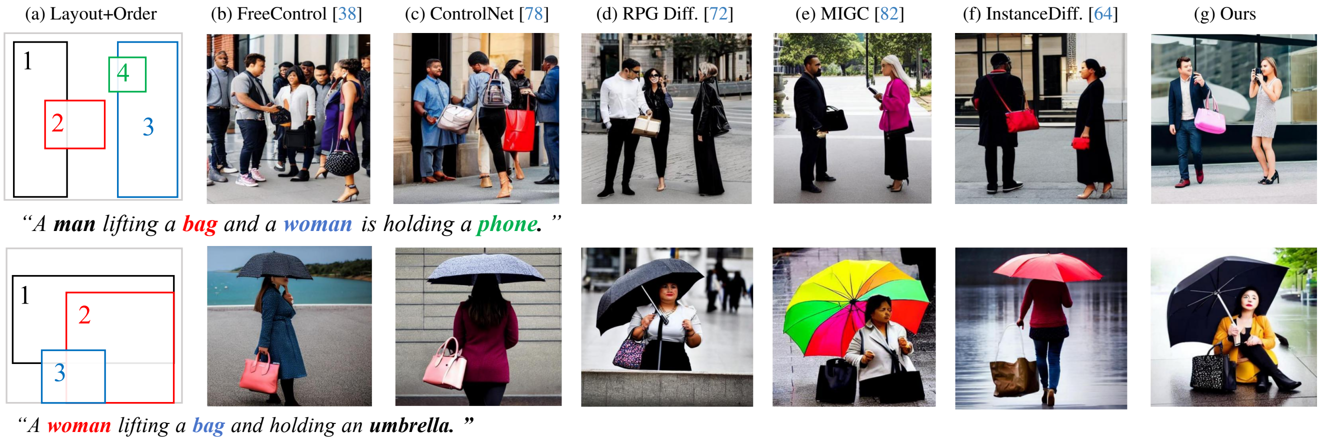
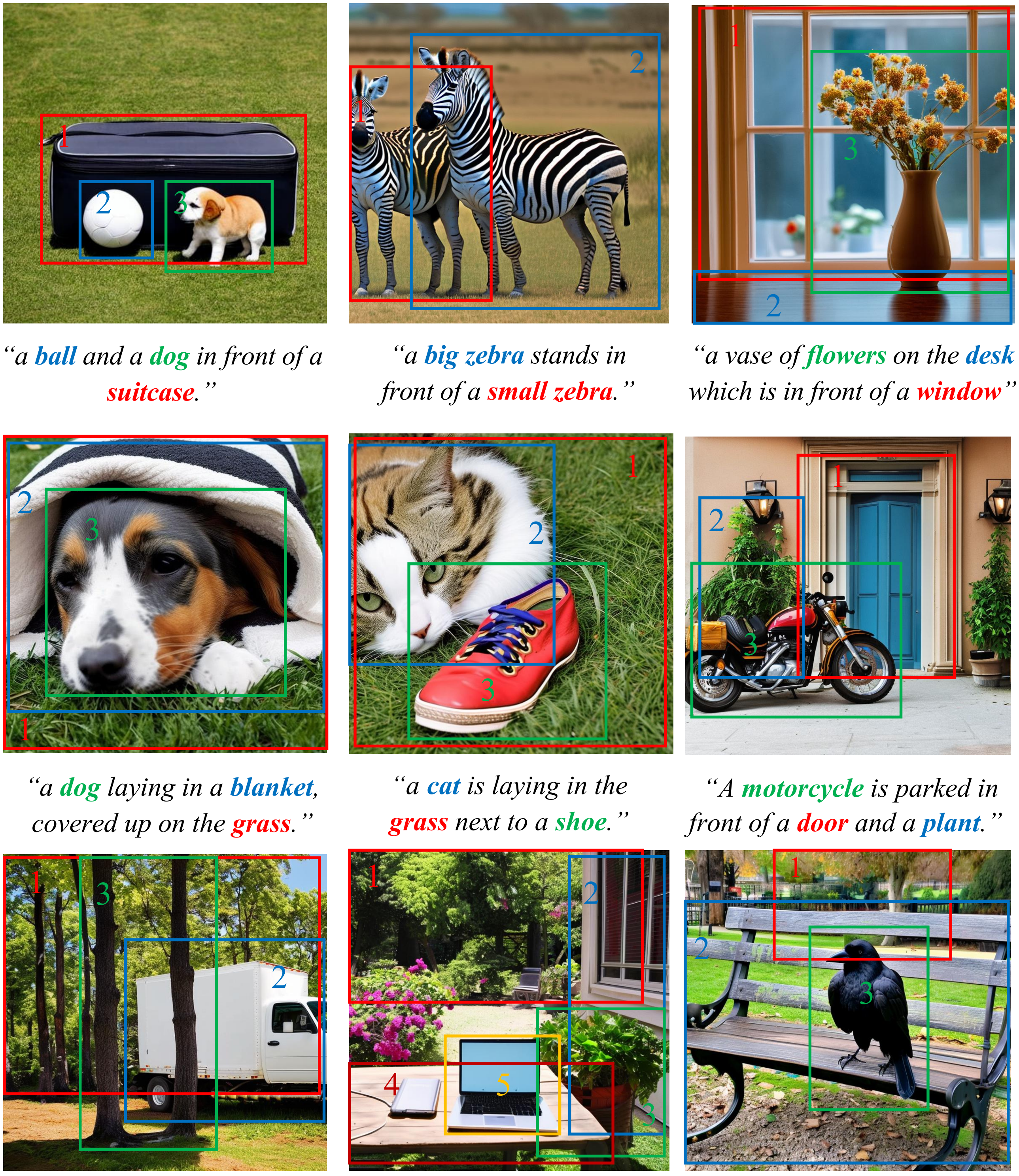

(a) Layout+Order
(b) FreeControl [38]
(c) ControlNet [78]
(d) RPG Diff. [72]
(e) MIGC [82]
(f) InstanceDiff. [64]
(g) Ours
1
4
2
3
"A **man** lifting a **bag** and a **woman** is holding a **phone**. "
1
2
3
"A **woman** lifting a **bag** and holding an **umbrella**. "

Figure 8. Results of cases with different visual orders for different parts of the area of the object.



*“a **ball** and a **dog** in front of a **suitcase**.”*

*“a **big zebra** stands in front of a **small zebra**.”*

*“a vase of **flowers** on the **desk** which is in front of a **window**”*

*“a **dog** laying in a **blanket**, covered up on the **grass**.”*

*“a **cat** is laying in the **grass** next to a **shoe**.”*

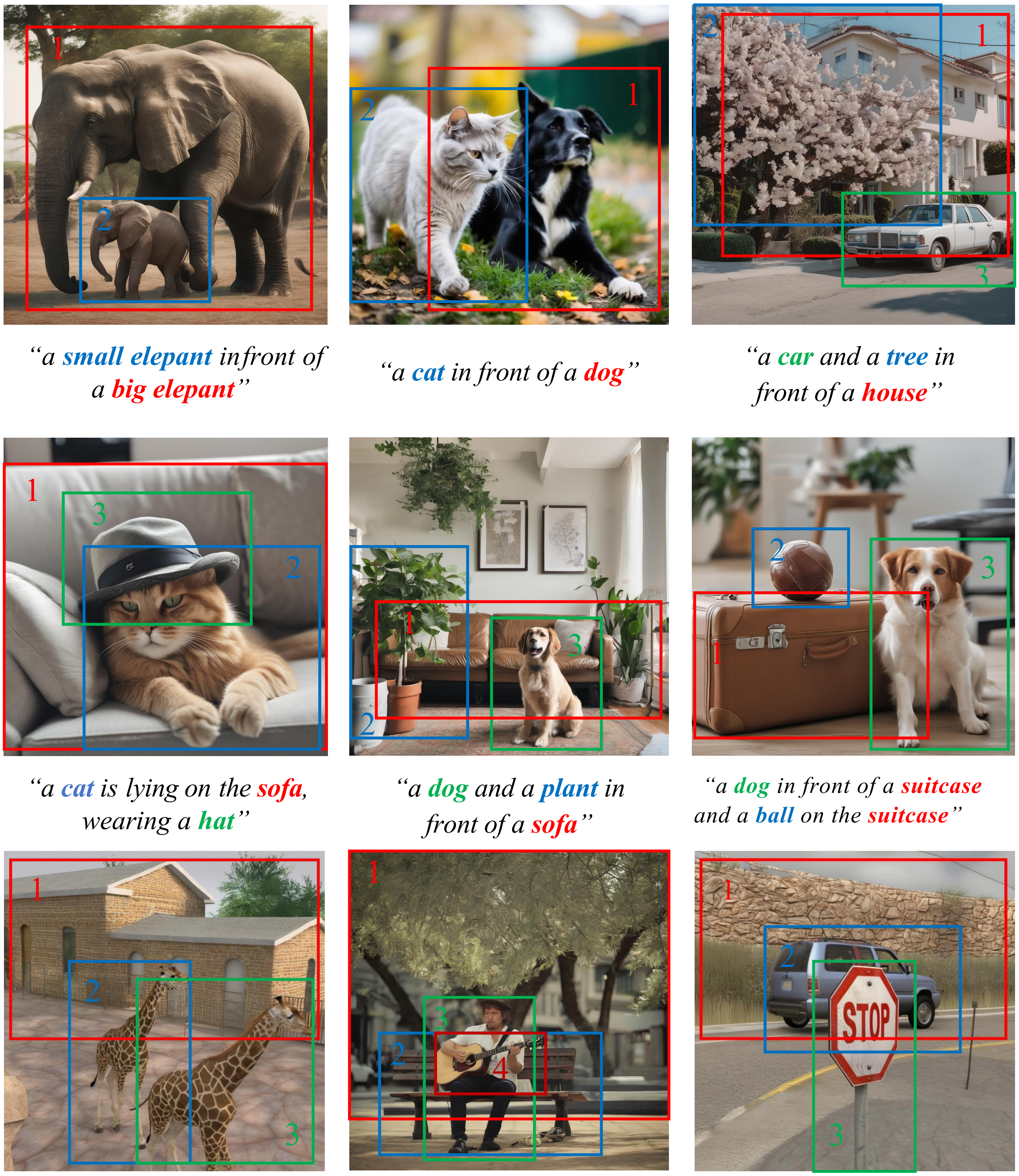
*“A **motorcycle** is parked in front of a **door** and a **plant**.”*

*“A **white truck** parked in front of **trees** but behind **two large trunks**. ”*

*"A **laptop** is on a **table** in front of **trees** and a **plant**,which is beside a **house**. "*

*"a **black bird** on a **bench** which is in front of a **tree** "*

Figure 9. More Results of our method using SD 1.5



“a ***small elepant*** *infront of a* ***big elepant****”*

“a ***cat*** *in front of a* ***dog****”*

“a ***car*** *and a* ***tree*** *in front of a* ***house****”*

“a ***cat*** *is lying on the* ***sofa****, wearing a* ***hat****”*

“a ***dog*** *and a* ***plant*** *in front of a* ***sofa****”*

“a ***dog*** *in front of a* ***suitcase*** *and a* ***ball*** *on the* ***suitcase****”*

“A ***giraffe*** *is in front of* ***another giraffe****, and they are both in front of the* ***houses****”*

"A ***man*** *sitting on a* ***bench*** *in front ofa* ***tree****, playing a* ***guitar****. "*

"A ***stop sign*** *is in front of a* ***car****, and the car is in front of a* ***wall****. "*

Figure 10. More Results of our method using SDXL.

pute the masks for the regions of the new objects (M_i^{visible}, M_i^{overlap}, $M_i^{\text{background}}$). Subsequently, our SDP applies visual guidance to denoise the objects stage by stage according to a user-defined visual sequence.

Unlike directly generating an image from a layout, object insertion requires preserving the content of existing objects in the original image, with changes limited to the newly inserted objects. During the stages corresponding to existing objects, the noise representations of these objects and the background regions are constrained to align with the noise representation of the original image, obtained through DDIM inversion [58]. For regions corresponding to the new objects (M_i^{visible}), weaker visual guidance is applied initially. In the stages for the newly inserted objects, stronger visual guidance is applied. Simultaneously, the VOA loss is utilized to optimize the cross-attention map for the newly added objects, further enhancing the accuracy of generated occlusion relationships and object positioning. Finally, during the global denoising phase, consistency between the background and other object regions with the original image is enforced to ensure seamless integration of the newly inserted objects.

As shown in Fig. 12, we demonstrate the application of our method for inserting newly generated objects into an existing image. In the first row of the figure, we showcase the insertion of new objects either in front of or behind existing objects in the original image. The second and third rows illustrate the insertion of multiple objects into the background while adhering to the user-specified visibility order among the objects. The inserted objects respect the predefined visibility order relative to the original objects in the image, while the background remains unchanged.

Exemplar based image editing. Similar to other exemplar-based image editing methods [9, 61, 71], our approach integrates user-specified objects into the generated image while preserving their identities and background unchanged. However, our method uniquely enables control over the visual order of these objects, as demonstrated in Fig. 13.

To achieve identity preservation, we utilize DreamBooth [55] to learn the concepts of the objects in the input image, representing them as multiple text tokens. These learned text tokens, along with their corresponding bounding boxes, are used as input prompts for our *VODiff*, allowing the generated objects to not only preserve the identities of the input objects but also ensure that their positions and visibility order align with the preset arrangement.

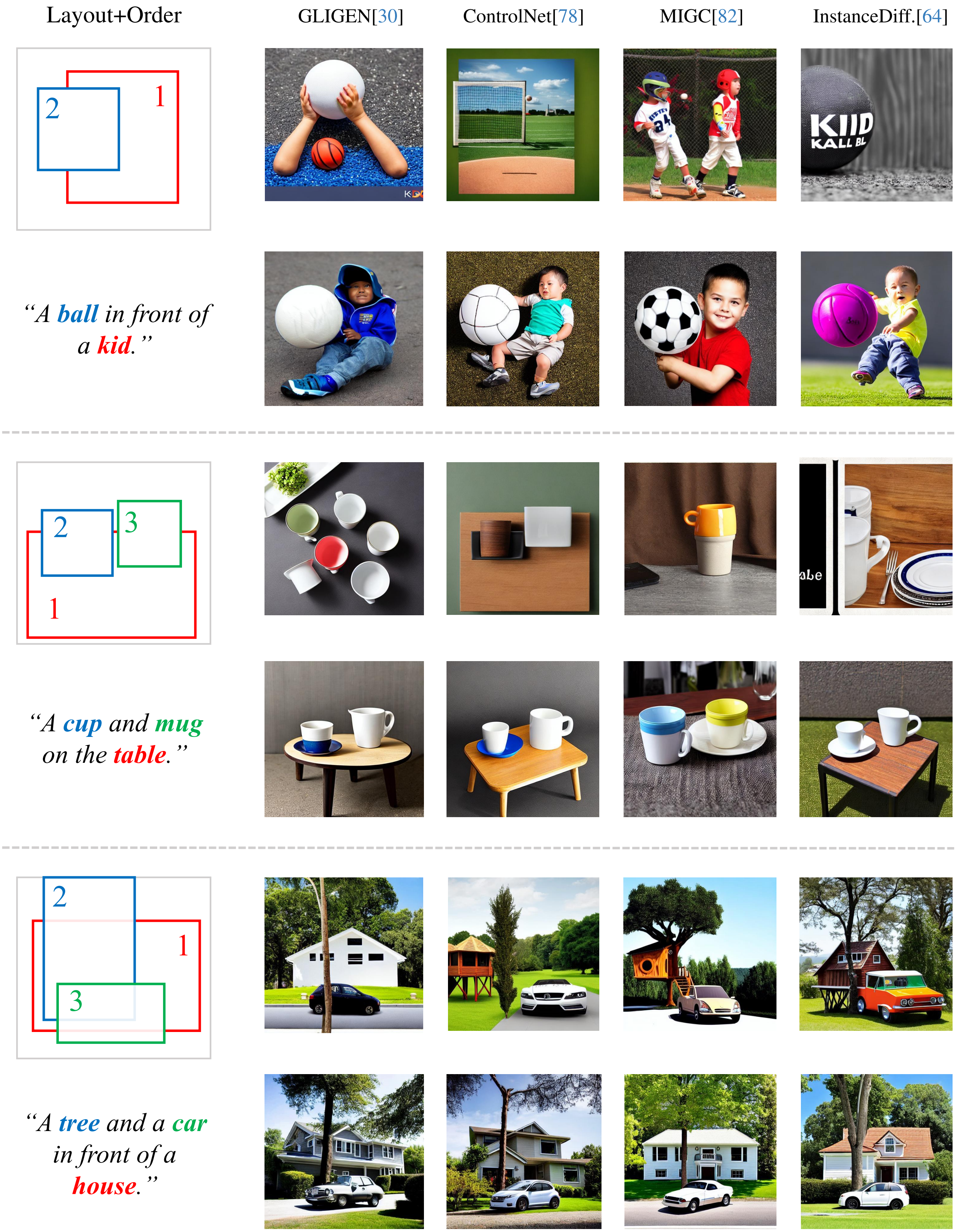

Layout+Order
GLIGEN[30]
ControlNet[78]
MIGC[82]
InstanceDiff.[64]
2
1
"A ball in front of a kid."
2
3
1
"A cup and mug on the table."
2
1
3
"A tree and a car in front of a house."

Figure 11. Enhancement of existing methods using *VODiff*. For each case, row 1 illustrates the baseline results, whereas row 2 demonstrates the enhanced outcomes achieved with our *VODiff*.

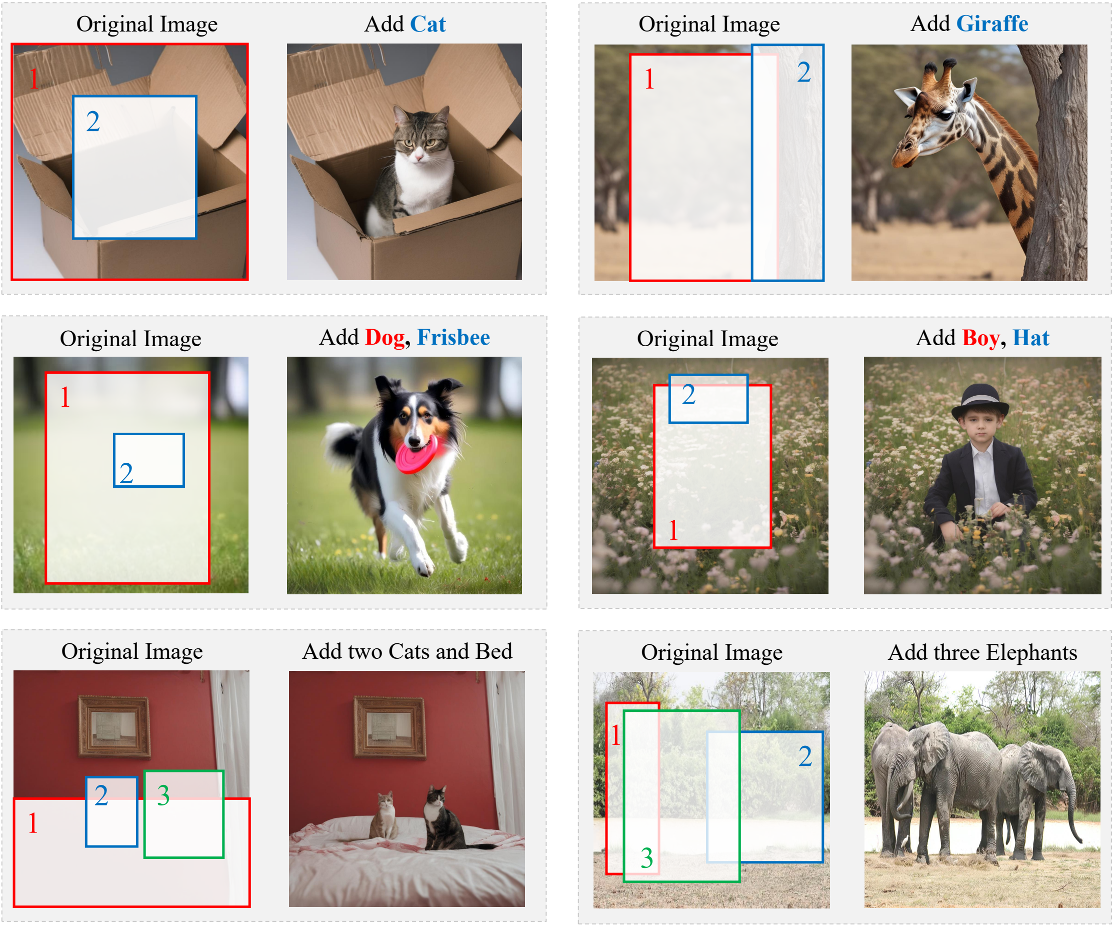


Figure 12. Object insertion with correct visibility orders.

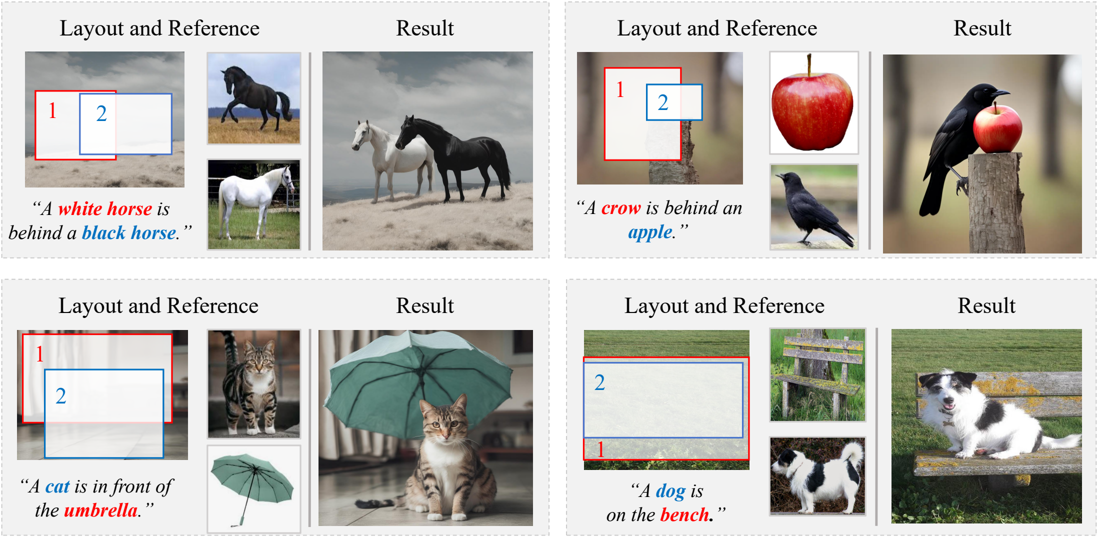


Figure 13. Exempler based image editing with accurate visibility orders.