This supplementary document offers additional comparisons and detailed information regarding our proposed RRNet model. In Sec A, we offer a further comparison between proposed RRNet and more baseline methods. In Sec B, we present more visualized results. We then provide additional analysis of V_{eot} in Sec C. The implementation and additional experimental details are included in Sec D. Finally, we discuss the limitations of RRNet in Sec E.

A. Additional Comparisons

Method	GPU Memory Usage (MiB)
TI	25458
Unet Update	30060
Unet Update + TI	OOM
RRNet	25040

Table 4. **Memory usage of training.**

Additional baselines. To provide a more comprehensive comparison, we compare with three other baseline methods.

- **Textual Inversion.** The first method is a variation of textual inversion [8]. In this approach, we employ the its loss function to fit the diffusion model to the given set of images by optimizing the input embeddings.
- **Unet Update.** The second method involves maintaining the text encoder of SD fixed while fine-tuning the Unet component. Specifically, we implement Low-Rank Adaptation (LoRA) [16] to prevent catastrophic forgetting in SD. The training parameters are set with a learning rate of 1e-4 and a rank of 4, over a duration of 100 epochs.
- **Reversion.** Furthermore, we tested the Reversion method on our dataset. The empirical results indicated that the images generated through Reversion were of poor quality. This suboptimal performance could be attributed to the inherent complexities associated with the Relation Rectification Task. Due to these findings, we decided against including the quantitative results of this method in our report. For reference, examples of these generated images are depicted in Figure 8.

A potential fourth method, optimizing the text encoder and Unet parameters simultaneously, was considered but deemed impractical due to its excessive computational demands. The memory requirements for training using different methods are compared in Table 4. Training with a batch size of 1 is not unfeasible on a single V100-32GB.

Quantitative analysis. The results are presented in the Table 5. It can be observed that among the baselines, TI demonstrates higher performance in relationship generation accuracy metrics compared to Unet Update. Our RRNet, employing a mere $\lambda = 0.2$, closely approaches the performance of TI across various relationship generation accuracy metrics, while also maintaining a lower FID. Increasing λ to 0.6 significantly elevates RRNet's performance above the baselines in all relationship generation accuracy metrics. Specifically, in the Position (Qwen) and Action (Qwen) metrics, RRNet (λ=0.6) outperforms TI by 13.7% and 5.7%, respectively. Furthermore, when using LLaVA as the detector, RRNet (λ=0.6) also exceeds TI by 9.2% and 5.8% respectively in position and action relationship generation metrics. In terms of OGA, RRNet (λ=0.6) surpasses TI by a margin of 4.1%.

For those two baselines, since the issue with the text encoder treating sentences as a Bag of Words (BOW) remains unresolved, they still struggle to accurately distinguish the direction of relationships. They tend more to directly fit the dataset rather than truly learning to represent the relationships between objects.

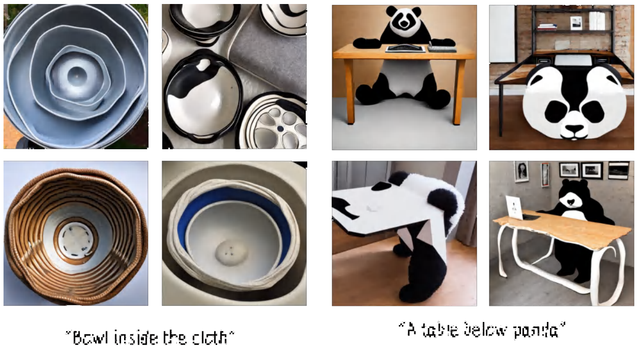


Figure 8. In our dataset, the Reversion fails to achieve accurate relationship generation.

B. Results Visualization

We provide additional qualitative results to demonstrate the effect of RRNet on relation rectification for SD. The generation results of action OSPs are shown in Figure 9 and the results of positional OSPs are showcased in Figure 10.

It can be observed that with RRNet's assistance, SD can accurately generate relationships in both directions.

C. Additional Analysis of V_{eot}

In the Sec 3, we identify the V_{eot} as pivotal in generating relationships. Specifically, as detailed in previous work [3], the V_{eot} plays a crucial role in foreground generation, encompassing both the objects and their interrelationships.

Here, we conducted an experiment by replacing the V_{eot} of two prompts to validate its significance.

Method	Position(Qwen) ↑	Position(LLaVA)↑	Action(Qwen) ↑	Action(LLaVA)↑	OGA↑	FID↓
TI	0.560	0.592	0.443	0.574	0.929	91.634
Unet Update	0.473	0.563	0.404	0.529	0.912	87.294
RRNet (λ=0.2)	0.564	0.597	0.469	0.565	0.937	89.13
RRNet (λ=0.6)	**0.697**	**0.684**	**0.500**	**0.632**	**0.970**	100.78

Table 5. **Further comparisons.**

Observation. Our experiment, illustrated in the first row of Figure 11, involved replacing the V_{eot} from "`A corgi`" with that from "`A cat inside the box`". This resulted in an image of "`A corgi inside the box`". This experiment suggests that the primary role of V_{eot} is to control the foreground layout, dictating where each object appears. Since the semantic of "`A corgi`" closely match that of "`A cat`", it would match the anchor for "`A cat`" and instead generate a corgi.

Discussion. We uncovered a fascinating aspect of V_{eot} related to the diffusion generation mechanism. Analogous to painting, the V_{eot} shapes the basic layout of the foreground, setting anchor points for each object's generation. It establishes an anchor point for each foreground object to be generated, without detailing each object specifically. The word embeddings of these objects then align with the closest anchor, leading to their manifestation at specific locations. The occurrences in Figure [Mask Phenomenon] can be seen as a failure in object generation due to the lack of precise positional information in V_{eot}.

Therefore, in our work, we achieve the effect of correctly controlling the generation of the foreground by adjusting V_{eot}.

D. Experiment Settings

D.1. Detailed dataset statistics.

The RR dataset comprises 21 relationships, consisting of 8 positional and 13 action types. Each relationship includes 4 types of prompts, two for OSPs represented as $< A, R, B >$ and $< B, R, A >$, and two generated from template sentence "`This is a photo of {obj}`" for object disentanglement purposes. We collect 3-5 images for each prompt to serve as exemplars for training.

D.2. Relationship Detection with Chatbots.

Prompts for relationship detection. We employ Vision-language chatbots to facilitate the detection of relationships in images. To ensure these chatbots focus more on the relationships between objects, we have developed a series of prompt templates. For a sentence can be abstracted as triplet $< A, R, B >$, prompt templates are listed follows:

1. "`Is there any object A in the image?`"
2. "`Is there any object B in the image?`"
3. "`Are both object A and object B present in the image?`"
4. "`Can you infer the relationship that exists between` *`object A`* `and` *`object`* `B in the image?`"
5. "`Is there ARB or BRA or neither?`"

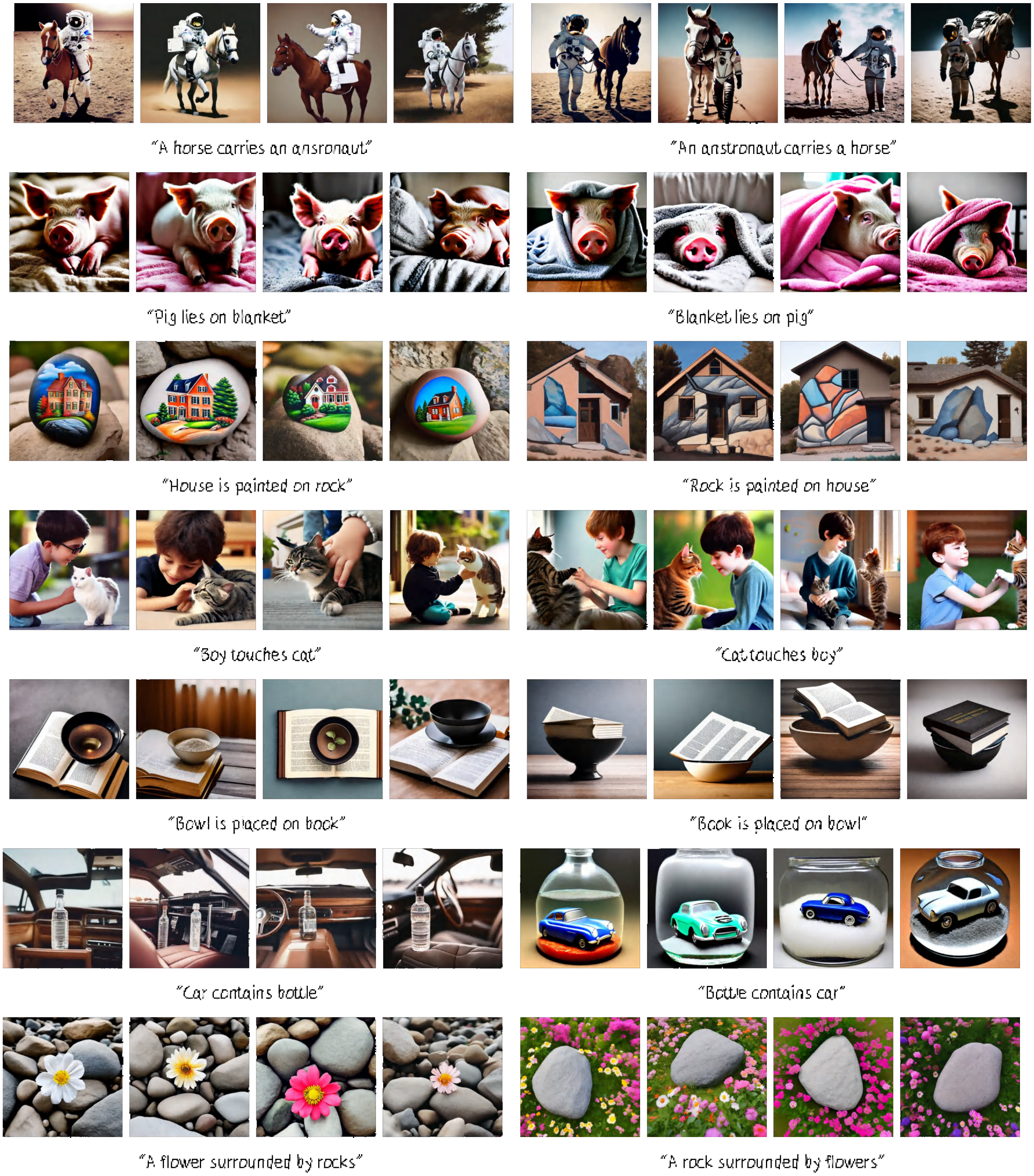
"`object A`" and "`object B`" are placeholders substituted by entities in the OSPs. "`ARB`" and "`BRA`" are OSPs. During the evaluation, each generated image undergoes assessment through above five questions.

Object detection. Question 1 and Question 2 are designed for object detection. During our experiments, we observed a high false-positive rate in object detection using the first two questions, primarily because the model occasionally generates a composite object embodying features of both objects A and B. Consequently, even if there is a single object displaying both sets of features, the chatbots are likely to affirmatively respond to the first two questions. To mitigate this issue, we introduced Question 3, aimed at filtering out objects that exhibit mixed features. For object detection, a generation is deemed correct only if it successfully passes Questions 1, 2, and 3.

Relationship detection. Question 4 is designed to steer the chatbot towards creating contexts that emphasize the relationships between objects, thereby setting the stage for Question 5. Regarding relationship generation, a relationship generation is classified as correct only if the chatbot's response to the Question 5 aligns with the prompt for generating the given image.

D.3. Additional Implementation details.

Implementation details of HGCN. In our implementation, the HGCN is built with DGL [43]. We use the HGCN based on Graph Attention Networks (GAT) [30]. The dimension of HGCN's hidden layers is 512. The learning rate of HGCN is set to 3e-4. The optimizer used is AdamW [31]. The training batch size is set to 1.



"A horse carries an ansronaut"

"An anstronaut carries a horse"

"Pig lies on blanket"

"Blanket lies on pig"

"House is painted on rock"

"Rock is painted on house"

"Boy touches cat"

"Cat touches boy"

"Bowl is placed on book"

"Book is placed on bowl"

"Car contains bottle"

"Bottle contains car"

"A flower surrounded by rocks"

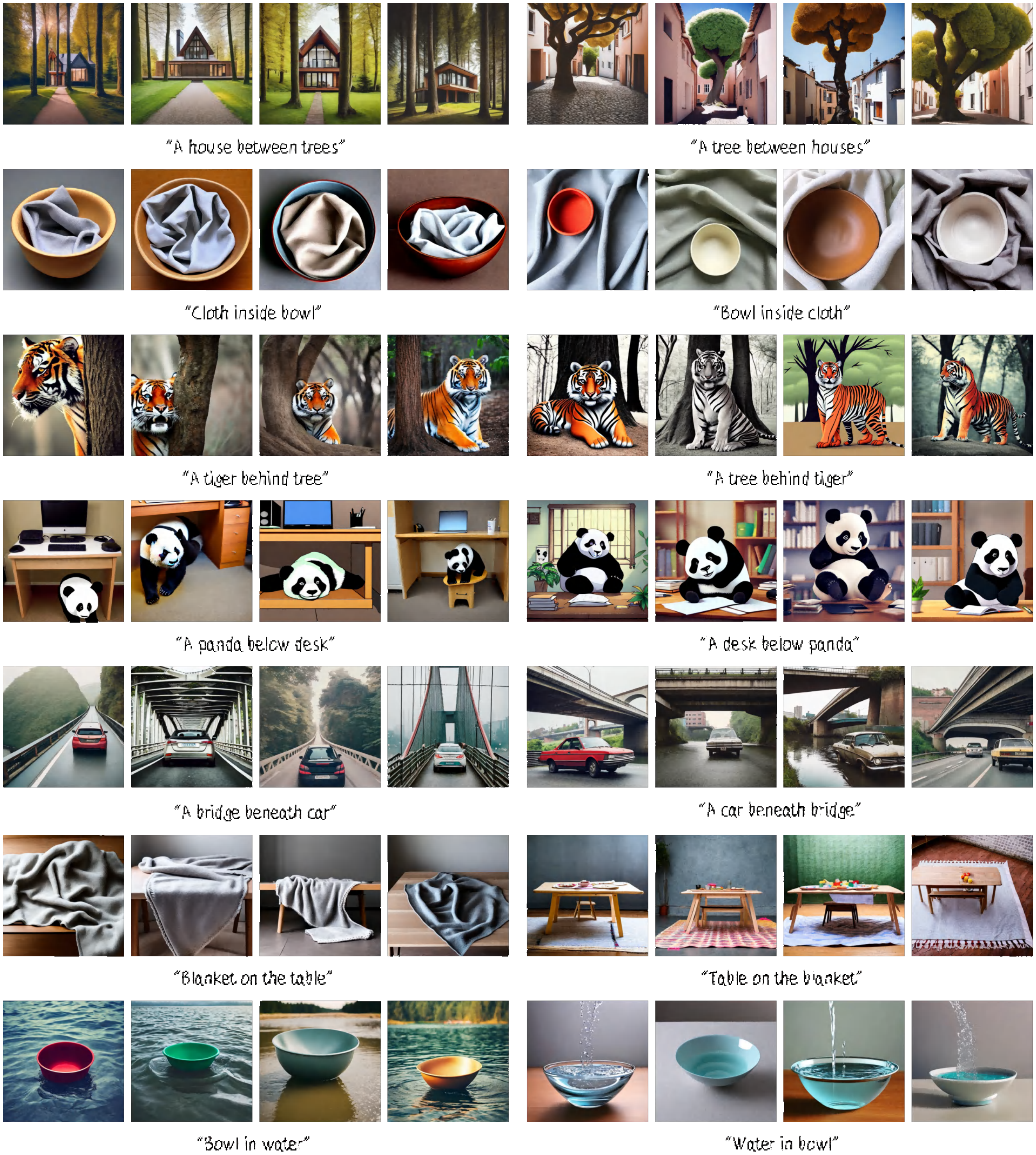
"A rock surrounded by flowers"

Figure 9. Additional results of action relation rectification with RRNet.

E. Limitations

E.1. Unseen concepts

Our method is capable of steer the generation of SD by adjusting the direction of the relations in the text


"A house between trees"

"A tree between houses"

"Cloth inside bowl"

"Bowl inside cloth"

"A tiger behind tree"

"A tree behind tiger"

"A panda below desk"

"A desk below panda"

"A bridge beneath car"

"A car beneath bridge"

"Blanket on the table"

"Table on the blanket"

"Bowl in water"

"Water in bowl"

Figure 10. Additional results of positional relation rectification with RRNet.

embeddings. Since we do not modify any parameters in denoising network, there are limitations to our approach for concepts that do not exist in SD. We show a failure case in Figure 12. Although our method separates the "`A horse rides an astronaut`" from "`An astronaut rides a horse`", SD lacks the

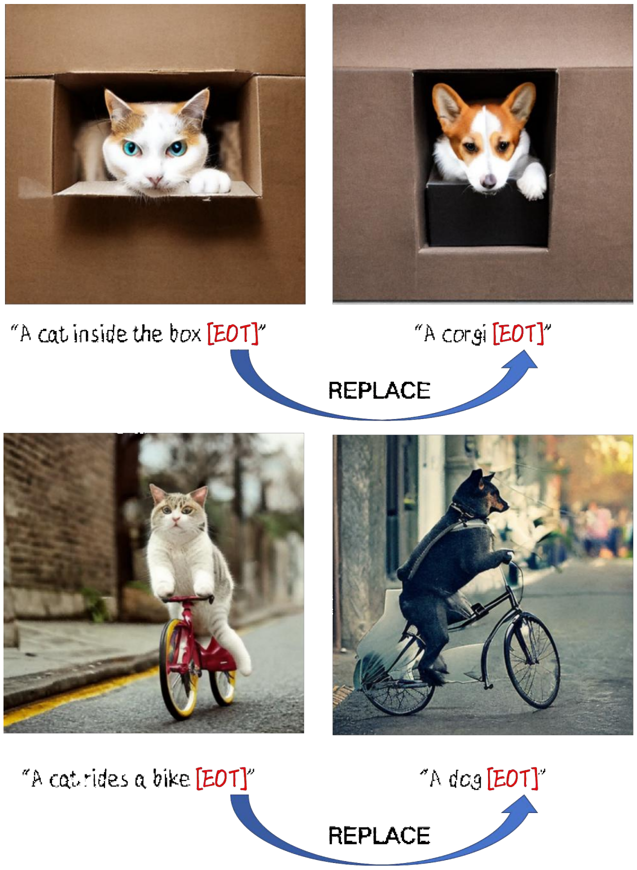


Figure 11. After replacing the V_{eot}, a phenomenon of entity replacement occurred.

concept of a horse riding anything, leaving the RRNet directionless in adjusting the horse-astronaut relationship. Consequently, most generated images are mere variations of existing dataset images. we observe that for such abstract relations, we need a larger λ to ensure the generated images meaningful, which in term undermines the images' quality.

E.2. Multi-relationships generation

we employed multiple RRNets, initially *trained on simple paired relations*, to handle more complex scenarios in image generation. The results are illustrated in the Figure 13. Although generated results looks reasonable, there is a noticeable drop in performance as the complexity of relationships and objects increases. The occurrence of this phenomenon is likely due to multiple RRNETs simultaneously adjusting the V_{eot}, resulting in semantic confusion within V_{eot}. Therefore, we believe that exploring how to construct a more complex graph to generate one adjustment vector capable of jointly rectifying multiple relational semantics is a highly promising avenue for future research.

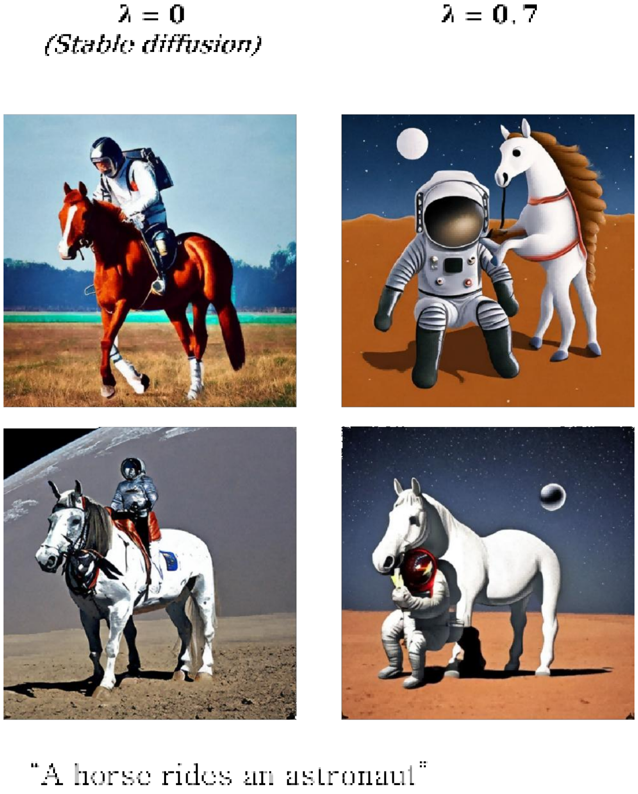


Figure 12. **Failure case.** The model has no sense of how horse rides on other objects.

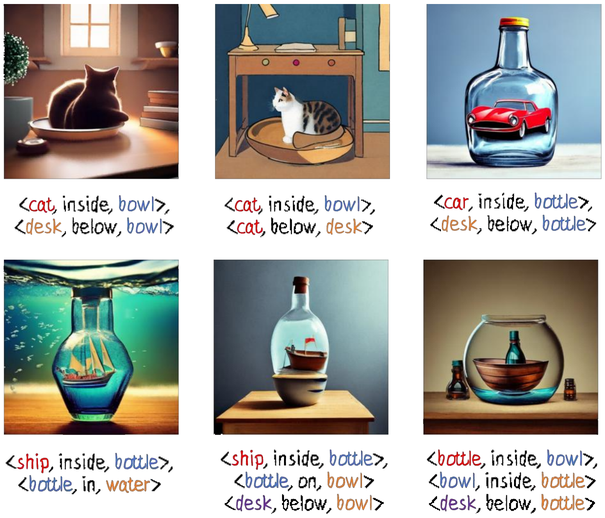


Figure 13. **Generation of Complex Relationships.**