

---

# Supplementary Material for *TriggerCraft*: A Framework for Enabling Scalable Physical Backdoor Dataset Generation with Generative Models

---

Anonymous Author(s)

Affiliation

Address

email

1 This supplementary material provides additional details and experimental results to support the main  
2 submission. We begin by providing experimental details in Sec. 1 and additional details about the  
3 devices in our physical evaluation of the poisoned models in Sec. 2. Then we provide the details  
4 of the real datasets in Sec. 3. We also conduct a human evaluation test for the Trigger Suggestion  
5 Module in Sec. 4. Next, we present qualitative results of the Poison Selection Module in Sec. 5, and  
6 additional Grad-CAM analysis in Sec. 6 synthesized dataset to show the compatibility between the  
7 comparability between the synthesized and real physical-world data. Lastly, Sec. 7 shows additional  
8 results generated/edited by our Trigger Generation Module.

## 9 1 Experimental Setup

10 To simulate a challenging real-world scenario, we select a 5-class subset of ImageNet [1], which  
11 consists of various general objects and animals, including *bags*, *bottles*, *chairs*, *dogs* and *cats*. These  
12 classes, all superclasses in ImageNet, are deliberately chosen to evaluate the effectiveness of our  
13 framework. This design choice emphasizes the inherent difficulty of identifying a common trigger  
14 object across diverse high-level categories, thereby demonstrating the robustness of our method under  
15 challenging conditions.

16 For the classifier, we select ResNet-18 [2] and employ SGD [3] as the optimizer, with a momentum  
17 of 0.9. The learning rate is set to 0.01 and follows a cosine learning rate schedule. We set the weight  
18 decay to  $1e-4$ , batch size to 64, and trained the model for 200 epochs. The default attack target is set  
19 to class 0 (*i.e.* dog). We employ a standard ImageNet augmentation from timm [4], with an input  
20 size of 224. All experiments are conducted on a server equipped with AMD EPYC 7513 32-Core  
21 processor and 7 Nvidia RTX A5000 24GB.

22 We utilize SG161222/Realistic\_Vision\_V5.1\_noVAE and InstructDiffusion as Image Gen-  
23 eration and Image Editing models, respectively.

## 24 2 Devices Used

25 In this section, we list the devices used to capture the real-world physical dataset, as detailed below:

- 26 • Huawei Y9 Prime 2019
- 27 • Xiaomi 11 Lite 5G
- 28 • Samsung M51
- 29 • Samsung Z Flip
- 30 • Realme RMX3263
- 31 • iPhone 13 Pro

- iPhone 15 Pro Max
- Ricoh GRIIIx camera

### 3 Dataset Distribution

This section presents the distribution of the ImageNet-5 [1] and the real-world physical data collected using the devices listed in Section 2. The dataset distributions are shown in Table 1 and Table 2, respectively.

For Table 2, the descriptions of the dataset are depicted as follows:

- **ImageNet-5-Clean:** A clean dataset of real images.
- **ImageNet-5-Tennis:** A poisoned real dataset where main subjects are captured along with a tennis ball.
- **ImageNet-5-Book:** A poisoned real dataset where main subjects are captured along with books.

Table 1: Distribution of ImageNet-5.

Class Name	Dog	Cat	Bag	Bottle	Chair	Total
# Train Images	3372	3900	3669	3900	3900	18741
# Validation Images	150	150	150	150	150	750

Table 2: Distribution of real physical world data.

Class Name	Dog	Cat	Bag	Bottle	Chair	Total
<b>ImageNet-5-Clean</b>	89	64	34	54	91	332
<b>ImageNet-5-Tennis</b>	164	152	67	82	141	606
<b>ImageNet-5-Book</b>	45	75	57	59	56	238

### 4 Human Evaluation Test for Trigger Suggestion Module

To evaluate the effectiveness of our Trigger Suggestion Module, we conduct a human evaluation study. We begin by generating a pool of 15 trigger objects where 5 are selected based on the suggestions from our Trigger Suggestion Module, while the remaining 10 are randomly generated. We randomly select a pool of 20 images and associate each image with a list of potential trigger objects. Human evaluators are then asked to identify the top 5 (trigger) objects from the list that naturally fit within the context of each image. In total, we collect 120 responses, as summarized in Table 3. The results show that in 96% of the cases, our Trigger Suggestion Module produced at least one suggestion that matched a human selected trigger, demonstrating strong contextual alignment. Notably, the most frequent outcome observed in 38% of responses involved exactly 2 matched suggestions, indicating that nearly half of the module’s outputs aligned well with multiple human judgments. This not only shows frequent agreement but also high precision in the suggestions. These findings underscore the effectiveness and contextual relevance of our Trigger Suggestion Module across diverse image scenarios.

As a note, all the evaluators are paid with minimum wages accordingly and IRB approvals are acquired beforehand.

### 5 Qualitative and Quantitative Results of Poison Selection Module

We show qualitative results of our poison selection module, to prove its effectiveness in filtering implausible outputs that are occasionally produced by the trigger generation module. The results are shown in Fig. 5, 6, 7 and 8, respectively.

Additionally, we show the ImageReward [5] scores for both image editing and image generation models for “tennis ball” in Fig. 3 and “book” in Fig. 4. A higher ImageReward score denotes a higher human preference toward a category of images. Generally, generated images have higher

Table 3: Human Evaluation Test for Trigger Suggestion Module

# of Matched Human Suggestions	Count	Percentage	% of Matched VQA Suggestions
0	5	4%	100%
1	14	12%	96%
2	46	38%	84%
3	32	27%	46%
4	19	16%	19%
5	4	3%	3%

ImageReward scores compared to edited images. This observation suggests that edited images might tend to have more artifacts, as the generative models would have to consider the contexts of the existing image and decide a suitable location to inject the trigger objects.

## 6 Additional Grad-CAM Analysis

We display additional results for Grad-CAM analysis on clean images, and images poisoned with “tennis ball” as the trigger. As for the images poisoned with “tennis ball” in Fig. 2, we observe that the backdoored model focuses on the “tennis ball”, leading to a successful backdoor attack. Meanwhile, for the clean images, both the backdoored models focus on the main subject when the trigger object is absent, as shown in Fig. 1. Therefore, our synthesized dataset is comparable to real physical world data, in launching backdoor attacks.

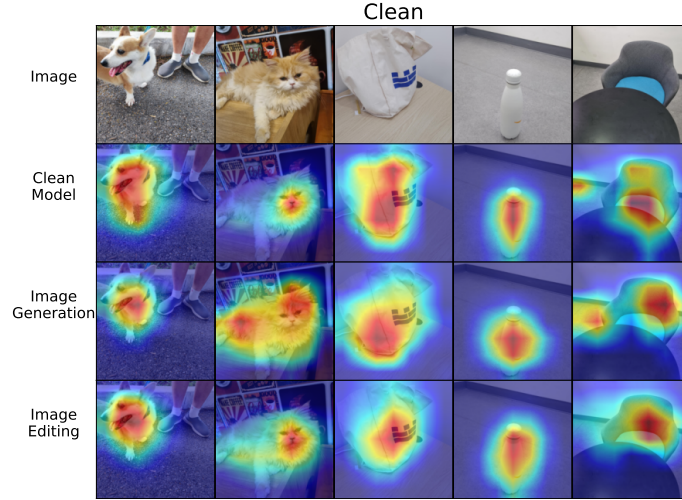


Figure 1: Grad-CAM of the clean model and backdoored model on clean real images, captured with multiple devices under various conditions.

## 7 Additional Examples

In this section, we show additional examples (Fig. 9, 10, 11, 12) for both Image Editing and Image Generation models, and for both of the physical triggers (book and tennis ball). For most of the examples shown in the figures, we observe that the trigger objects are present coherently with the main subject, which proves the efficacy of our framework in synthesizing physical backdoor datasets. Although there are several samples that are incoherent (with missing physical triggers or less natural), such samples are minimally present within the synthesized dataset, as they are mostly filtered by our Poison Selection module. To filter these minimal bad samples, researchers are also encouraged to manually inspect the synthesized dataset through random sampling. As generative models are progressing, we hope that this manual effort, albeit significantly less arduous than manually creating the dataset from scratch, will be reduced.

Also, all generated or edited images have been verified to contain no sensitive content.

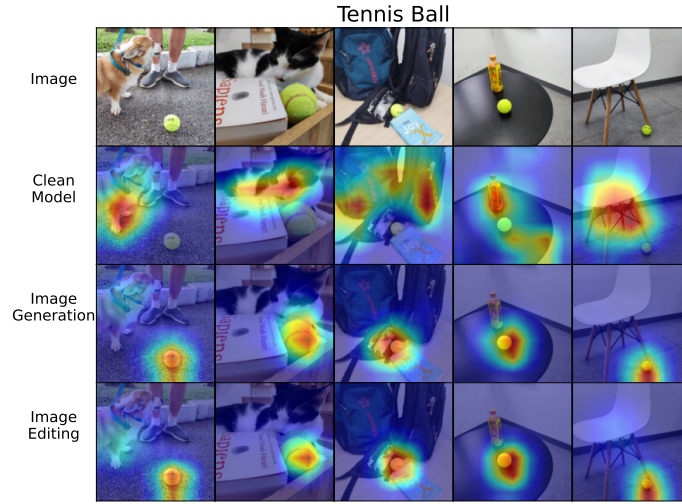


Figure 2: Grad-CAM of the clean model and backdoored model on real images with “tennis ball” as a trigger, captured with multiple devices under various conditions.

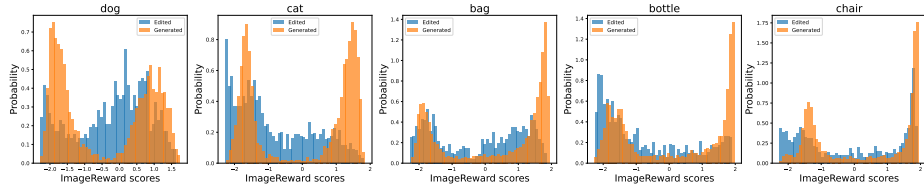


Figure 3: ImageReward scores for edited and generated images for the trigger - “tennis ball”.

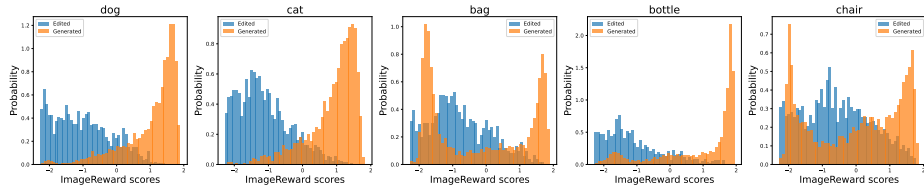


Figure 4: ImageReward scores for edited and generated images for the trigger - “book”.



Figure 5: Top and bottom *edited* images ranked by our poison selection module (ImageReward) for the trigger - “tennis ball”.





Figure 6: Top and bottom *edited* images ranked by our poison selection module (ImageReward) for the trigger - “book”.

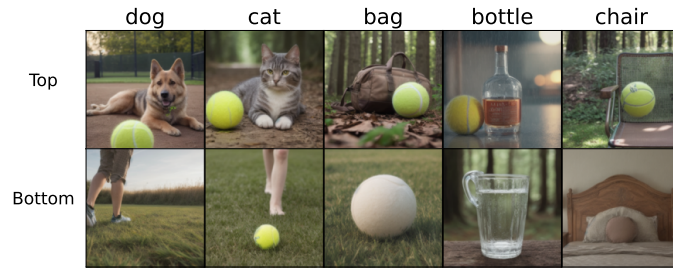


Figure 7: Top and bottom *generated* images ranked by our poison selection module (ImageReward) for the trigger - “tennis ball”.



Figure 8: Top and bottom *generated* images ranked by our poison selection module (ImageReward) for the trigger - “book”.

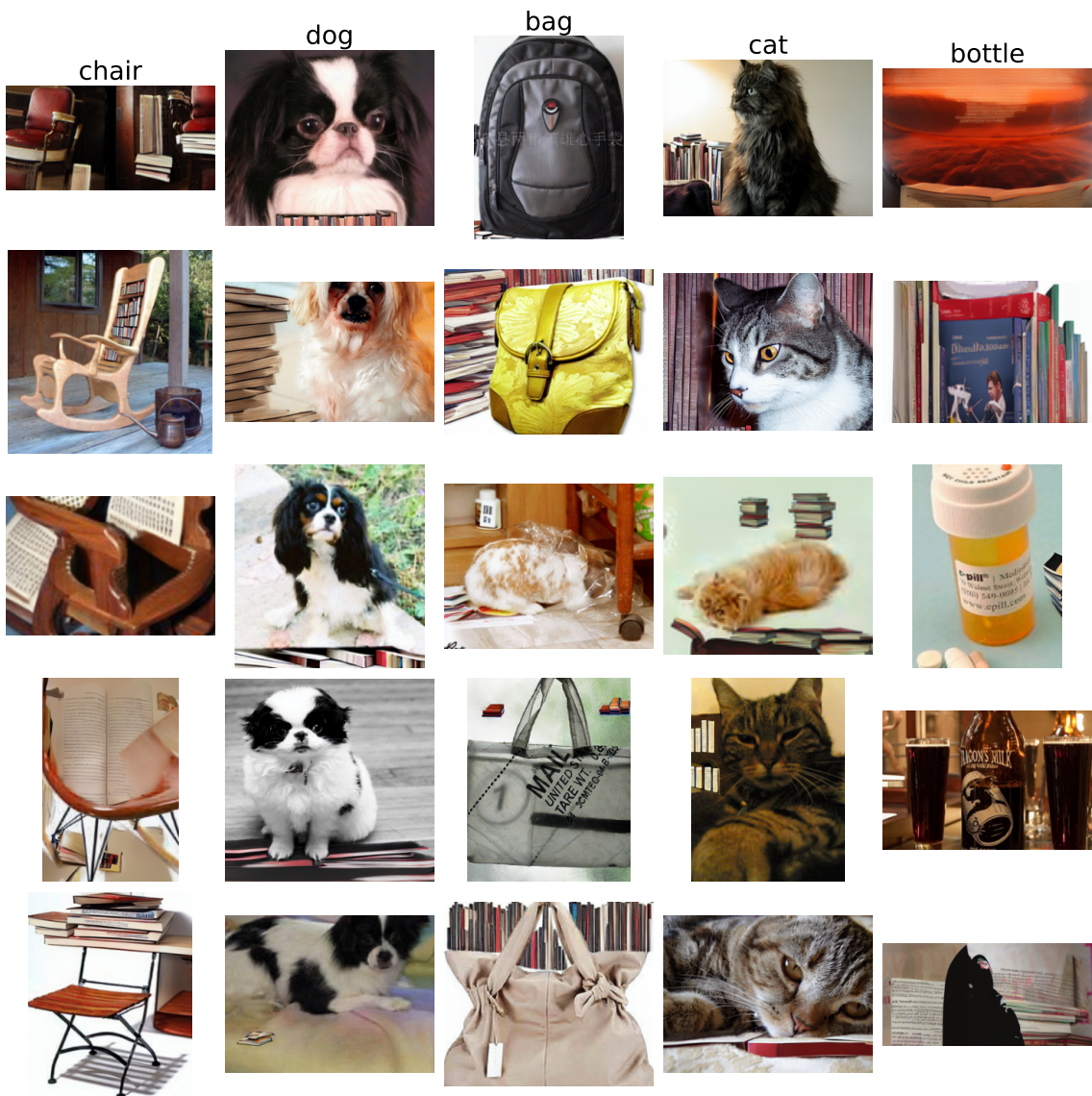


Figure 9: Additional examples of **edited images** for the trigger - “book”.

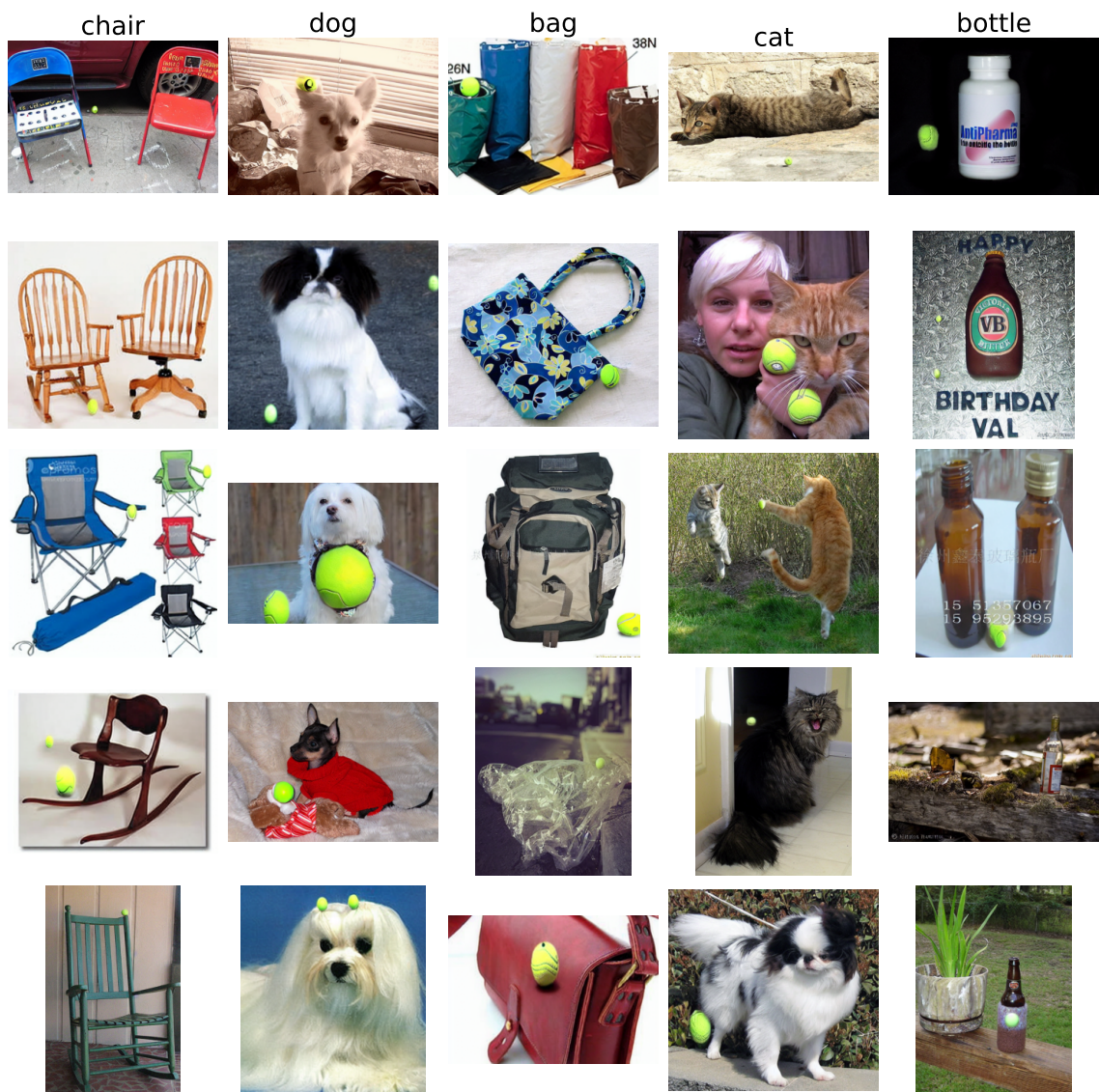


Figure 10: Additional examples of **edited images** for the trigger - “tennis ball”.



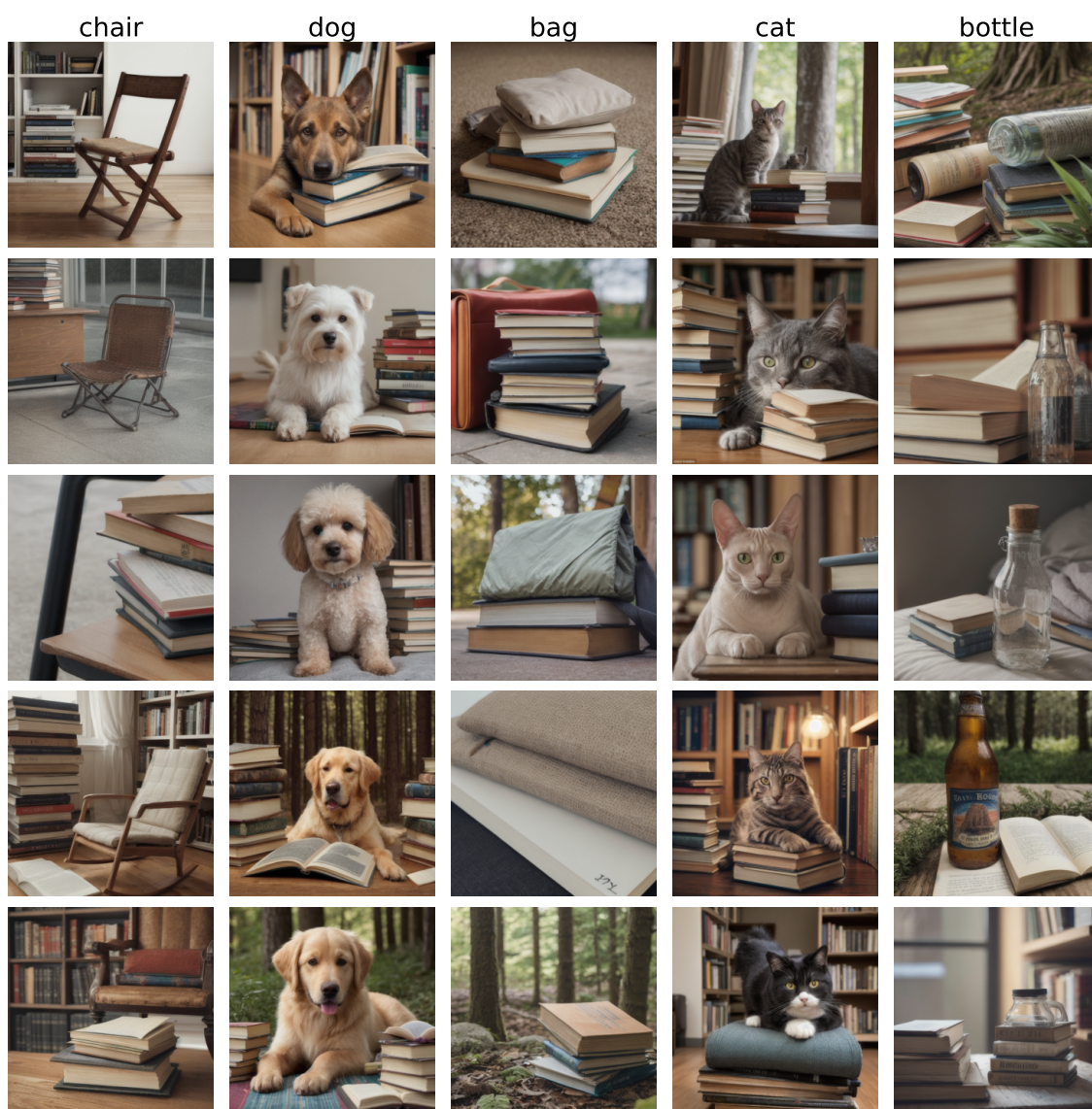


Figure 11: Additional examples of **generated images** for the trigger - “book”.

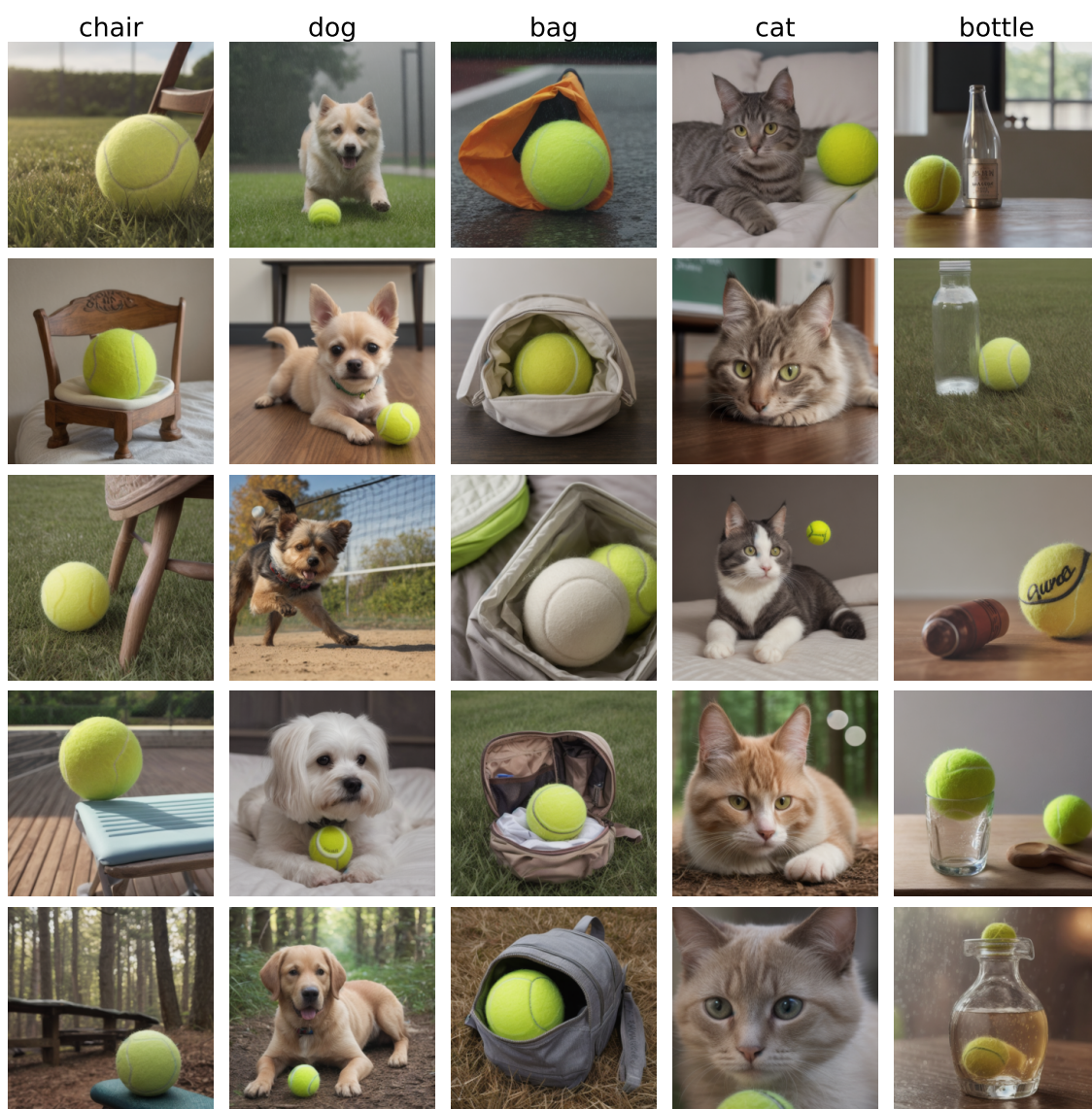


Figure 12: Additional examples of **generated images** for the trigger - “tennis ball”.

## 89 References

- 90 [1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale  
91 hierarchical image database. In *Proceedings of the 2009 IEEE Computer Society Conference on*  
92 *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, FL, 2009.
- 93 [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image  
94 recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern*  
95 *Recognition (CVPR)*, pages 770–778, Las Vegas, NV, 2016.
- 96 [3] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of*  
97 *Mathematical Statistics*, pages 400–407, 1951.
- 98 [4] Ross Wightman. Pytorch image models. [https://github.com/rwightman/](https://github.com/rwightman/pytorch-image-models)  
99 [pytorch-image-models](https://github.com/rwightman/pytorch-image-models), 2019.
- 100 [5] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao  
101 Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation.  
102 *Advances in Neural Information Processing Systems (NeurIPS)*, 36:15903–15935, 2023.