Appendix
Yuan: Yielding Unblemished Aesthetics through A Unified Network for Visual Imperfections Removal in Generated Images

Zhenyu Yu [1, *], **Chee Seng Chan** [1]

[1] Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, 524005, Malaysia
Corresponding author: yuzhenyuyxl@foxmail.com

Detailed Method

Automatic Mask Generation

Grounded SAM integrates the precise object detection capabilities of Grounding DINO with the segmentation strengths of the Segment Anything Model (SAM). This integration automates the generation of masks based on the synthesis prompt, effectively eliminating the need for manual intervention.

Grounding DINO is first employed for detailed object detection using a transformer-based architecture. Its loss function ($\mathcal{L}_{GDINO}$) includes both classification ($\mathcal{L}_{cls}$) and localization ($\mathcal{L}_{loc}$) components to ensure accuracy:

$$\mathcal{L}_{GDINO} = \sum_{i=1}^{N} \left[\mathcal{L}_{cls}(c_i, \hat{c}_i) + \alpha \cdot \mathcal{L}_{loc}(b_i, \hat{b}_i) \right] \quad \text{(A.1)}$$

Here, c_i and $\hat{c}_i$ represent the true and predicted class labels, while b_i and $\hat{b}_i$ are the true and predicted bounding boxes. The parameter α balances the importance of these terms. Grounding DINO's object detection outputs D_{GDINO} serve as a foundation for subsequent segmentation tasks.

Following object detection, the SAM model is applied to the identified regions to perform segmentation. The SAM model excels by learning from a vast and diverse dataset, optimizing a comprehensive loss function to achieve high segmentation accuracy:

$$\mathcal{L}_{SAM} = \sum_{i=1}^{N} \left[-y_i \log(\hat{y}_i) + \lambda \cdot \text{Reg}(\hat{y}_i) \right] \quad \text{(A.2)}$$

where y_i and $\hat{y}_i$ denote the true and predicted segmentations respectively, and λ is a regularization parameter. By conditioning on both the prompt and image features, Grounded SAM generates segmentation masks M_{SAM}:

$$M_{SAM} = \text{SAM}(I, D_{GDINO}) \quad \text{(A.3)}$$

This approach ensures robust and automated mask generation, addressing the limitations of manual methods and enhancing consistency and precision in identifying regions of interest.

Inpainting for Image Preservation

To preserve the original characteristics of the image, we adopt the LaMa Inpainting model over traditional diffusion-based methods. Diffusion-based techniques often introduce inconsistencies and artifacts that can detract from the image's coherence. In contrast, the LaMa model focuses on inpainting, which involves restoring specific masked regions based on the surrounding context.

The LaMa model is optimized to inpaint large masked regions effectively, predicting and filling these areas while maintaining visual consistency. This process is governed by a loss function that balances reconstruction and perceptual similarity:

$$I_{inpaint} = \text{LaMa}(I, M_{SAM}) \quad \text{(A.4)}$$

$$\mathcal{L}_{inpaint} = \mathcal{L}_{recon}(I, I_{inpaint}) + \beta \cdot \mathcal{L}_{perc}(I, I_{inpaint}) \quad \text{(A.5)}$$

where $\mathcal{L}_{recon}$ ensures the inpainted region matches the original image's appearance, and $\mathcal{L}_{perc}$ maintains high-level perceptual similarity. The parameter β controls the balance between these two objectives. This inpainting approach ensures that modified regions blend seamlessly with untouched areas, maintaining the original image's visual integrity and coherence.

Refining Visual Imperfects

Our approach to refining visual imperfections consists of two key steps: (1) adjusting the output logits of the SAM to obtain more accurate masks, and (2) employing Prompt-to-Prompt techniques for image repainting. If the results from step (1) are unsatisfactory, we proceed to step (2).

Adjusting Logits for Improved Masks Mask generation plays a crucial role in image processing, helping to identify and separate regions of interest within an image. Initially, masks were generated using a threshold value $t = 0$ to delineate mask boundaries. However, this threshold often failed to capture shadowed areas effectively, resulting in incomplete masks. To address this, we lowered the threshold t to achieve better feature coverage:

$$\Delta M_{SAM}(x) = \begin{cases} 1, & \text{if } logit(x) \geq t \\ 0, & \text{otherwise} \end{cases} \quad \text{(A.6)}$$

In our experiments, we achieved the best results by adjusting t. We have tentatively set the default threshold $t = -5$, as this value provides excellent results across multiple images. However, the optimal threshold can vary for different images due to the complexity of their features. Therefore, this parameter is user-adjustable, allowing for adapting to meet specific requirements. The adjusted threshold effectively improves mask coverage, especially for shadows and complex backgrounds.

Repainting via Prompt Instruct When the adjusted masks do not meet the desired level of refinement, we employ Prompt-to-Prompt techniques for further optimization. This technique guides the model to generate target images by analyzing semantic differences between the original and repainted images.

Caption can be generated using any caption generated model ($Caption$). To construct a high quality training dataset, we use Florence2 (Xiao et al. 2024) to generate captions for the images. It provide precise and detailed semantic information, which helps identify differences between the original and repainted images. Combined with user input modification requests (P) (e.g., `remove <the dog>`), we create a dataset for fine-tuning a GPT model to learn mappings between original (C_I) and inpainted captions (C_r), and then achieve caption instruct. Finally, we can use GPT$_{ft}$ to generate refined caption ΔC_r as the instruct prompt of any T2I generation models ($Generate$). In this paper, we use stable diffusion (SD) as an example.

$$\text{Caption}(I) \rightarrow C_I, \quad \text{Caption}(I_{inpaint}) \rightarrow C_r \tag{A.7}$$

$$GPT_{ft}(P, C_I) \rightarrow \Delta C_r \tag{A.8}$$

The fine-tuned GPT_{ft} learns to interpret and execute image optimization instructions, generating captions that serve as precise inputs for $Generate$ models. These captions guide SD in creating optimized images that meet expectations and enhance visual quality. By leveraging this Prompt-to-Prompt method, our approach significantly improves visual consistency and fidelity, particularly in scenarios that require extensive content modifications. The integration of GPT-generated captions with SD enables a seamless transition from textual instructions to high-quality visual outputs, ensuring that the final images align closely with user intents and aesthetic goals.

Through this two-step optimization strategy, our method demonstrates great flexibility and adaptability, offering precise visual refinement solutions tailored to different image features and user requirements.

Figures and Tables

This section includes:

(1) Effect of buffer zone thresholds (b) on reconstruction quality (Figure A.1);

(2) The result of our works (Figure A.2);

(3) Comparison of buffer zone thresholds (b) across datasets (Table A.1);

(4) Impact of $logits$ threshold (t) on auto-mask sensitivity and image quality (Table A.2).

Algorithm 1: *Yuan* - Object Removal

```
Require: Synthetic image $I$ from any T2I model
  Prompt $P$ from user input
Ensure: Refined image $I_{refined}$
  $D_{GDINO} \leftarrow$ GDINO$(I, P)$ {Detect objects}
  $M_{SAM} \leftarrow$ SAM$(D_{GDINO})$ {Generate mask}
  $I_{masked} \leftarrow$ Apply $\Delta M_{SAM}$ to $I$
  $I_{inpaint} \leftarrow$ LaMa$(I_{masked}, \Delta M_{SAM})$ {Inpaint}
  $output \leftarrow I_{inpaint}$
  if $I_{inpaint}$ is insufficient then
    $\Delta M_{SAM} \leftarrow logit(t)$ {Adjust mask}
    $\Delta I_{masked} \leftarrow$ Apply $\Delta M_{SAM}$ to $I$
    $I_{inpaint2} \leftarrow$ LaMa$(\Delta I_{masked}, \Delta M_{SAM})$ {Inpaint}
    $output \leftarrow I_{inpaint2}$
    if $I_{inpaint2}$ is insufficient then
      $C_I \leftarrow$ Caption$(I)$ {Generate caption}
      $C_r \leftarrow$ GPT$_{\text{fine-tuned}}(P, C_I)$ {Generate new caption}
      $I_{refined} \leftarrow$ Generate$(\Delta C_r, I)$
      $output \leftarrow I_{refined}$
    end if
  end if
  return $output$
```

Generated-cats Prompt List

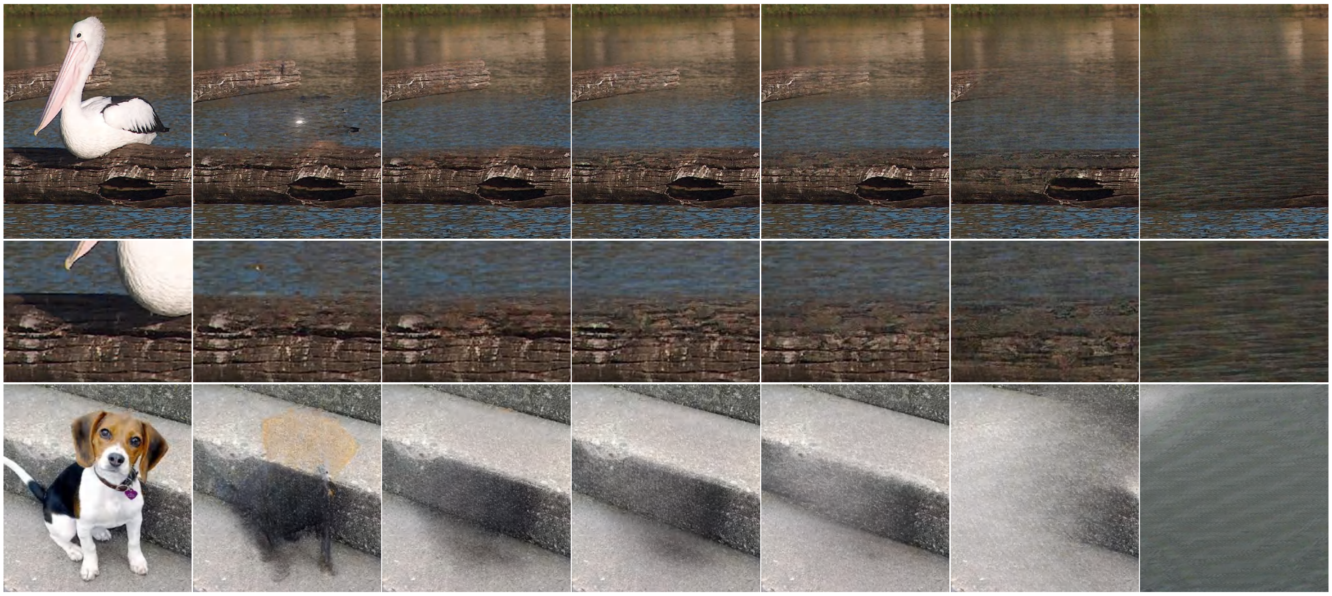
01. A regal black cat perched on a windowsill, silhouetted against a sunset.
02. A fluffy white kitten curled up in a cozy blanket.
03. Two playful tabby cats chasing each other through a field of flowers.
04. A sleek Siamese cat lounging on a velvet chaise longue.
05. A ginger cat peeking out from behind a stack of books.
06. A majestic Maine Coon cat sitting proudly in a sunbeam.
07. A mischievous calico cat batting at a dangling toy.
08. A sleepy Scottish Fold cat nestled in a pile of cushions.
09. A curious black and white tuxedo cat investigating a potted plant.
10. A fluffy Persian cat being groomed by its owner.
11. A playful kitten pawing at a butterfly.
12. A majestic Bengal cat prowling through tall grass.
13. A pair of cuddling kittens wrapped in a rainbow-colored blanket.
14. A contented cat lounging in a hammock.
15. A graceful cat leaping over a fence.
16. An elegant Russian Blue cat sitting by a window, watching the rain.
17. A fluffy Ragdoll cat sprawled out on a soft rug.
18. A sleek black cat with bright green eyes lounging on a piano.
19. A tabby cat peering curiously out of a flowerpot.
20. A playful kitten batting at falling leaves.
21. A fluffy Maine Coon cat perched on a tree branch.
22. A Siamese cat stretching out lazily in the sun.
23. A ginger cat napping in a sunbeam.
24. A curious kitten investigating a laptop keyboard.

Table A.1: Comparison of buffer zone thresholds (b) across datasets.

b	ImageNet100			Stanford-dogs			Generated-cats		
	NIQE↓	BRISQUE↓	PI↓	NIQE↓	BRISQUE↓	PI↓	NIQE↓	BRISQUE↓	PI↓
0	3.7537	**24.9772**	**2.5064**	4.5526	**20.8205**	**2.9659**	6.6916	43.7922	6.9720
15	3.7425	26.6525	2.5558	4.6187	25.1237	3.2685	**6.2217**	37.4372	**6.7089**
30	**3.5438**	25.9997	2.7441	**4.3036**	21.0217	3.1316	6.9695	36.8356	6.9524
50	5.7662	45.6513	4.6176	7.7046	48.8762	7.3630	7.8724	**35.2731**	7.1823
100	7.0501	47.5065	6.5582	7.7046	48.8762	7.3630	13.4853	37.7114	8.0294
200	7.0501	47.5065	6.5582	7.7046	48.8762	7.3630	13.4853	37.7114	8.0294

Table A.2: Impact of *logits* threshold (t) on auto-mask sensitivity and image quality.

t	ImageNet100			Stanford-dogs			Generated-cats		
	NIQE↓	BRISQUE↓	PI↓	NIQE↓	BRISQUE↓	PI↓	NIQE↓	BRISQUE↓	PI↓
0.0	**3.7350**	**25.5632**	2.5786	4.6179	**27.6525**	3.3228	6.5708	40.2131	6.9190
-5.0	3.8162	25.8107	**2.5403**	**4.5064**	28.3264	**3.2824**	**6.3903**	40.0700	**6.8134**
-10.0	4.0561	27.7858	2.8707	6.5758	38.1330	5.8810	6.9659	40.8078	7.0764
-15.0	7.0501	47.5065	6.5582	7.7046	48.8762	7.3630	13.4853	**37.7114**	8.0294



(a) Image (b) 0 (c) 15 (d) 30 (e) 50 (f) 100 (g) 200

Figure A.1: Effect of buffer zone thresholds (b) on reconstruction quality.

25. A fluffy white Persian cat sitting in a wicker basket.

26. A black and white tuxedo cat sitting on a bookshelf.

27. A pair of cats cuddled up together on a windowsill.

28. A playful kitten chasing its tail.

29. A regal Abyssinian cat surveying its kingdom from atop a fence.

30. A sleepy Scottish Fold cat nestled in a pile of autumn leaves.

31. A fluffy Siberian cat batting at falling snowflakes.

32. A majestic Norwegian Forest cat sitting among autumn foliage.

33. A sleek black cat perched on a fencepost, silhouetted against the moon.

34. A pair of Siamese kittens play-fighting.

35. A ginger cat peeking out from under a blanket.

36. A contented cat curled up in a sunbeam.

37. A playful kitten pouncing on a feather toy.

38. A regal British Shorthair cat sitting on a velvet cushion.

39. A fluffy Ragdoll cat being cuddled by its owner.

40. A sleek Bengal cat stalking through a field of tall grass.

41. A pair of cats grooming each other.

42. A curious kitten investigating a Christmas ornament.

43. A fluffy Maine Coon cat perched on a snowy branch.

44. A Siamese cat basking in the glow of a fireplace.

45. A ginger cat peering out from behind a stack of presents.

46. A playful kitten batting at a string of lights.

47. A majestic Persian cat sitting on a windowsill, framed by lace curtains.

48. A black and white tuxedo cat curled up in a cozy armchair.

49. A pair of cats napping together on a sunny porch.

50. A regal Abyssinian cat perched on a stone wall, gazing into the distance.

51. A fluffy Siberian cat chasing after falling cherry blossoms.

52. A sleek black cat prowling through a garden at night.

53. A pair of Siamese kittens cuddled up together in a basket.

54. A ginger cat playing with a ball of yarn.

55. A contented cat lounging in a sun-dappled patch of grass.

56. A playful kitten batting at a toy mouse.

57. A regal British Shorthair cat sitting on a velvet throne.

58. A fluffy Ragdoll cat being carried in its owner's arms.

59. A sleek Bengal cat stalking through a field of wildflowers.

60. A pair of cats sitting side by side, staring out a window.

61. A curious kitten peeking out from inside a paper bag.

62. A fluffy Maine Coon cat perched on a fencepost.

63. A Siamese cat curled up in a cozy basket, surrounded by pillows.

64. A ginger cat gazing up at a butterfly.

65. A playful kitten chasing after a laser pointer.

66. A majestic Persian cat lounging on a luxurious velvet sofa.

67. A black and white tuxedo cat sitting on a windowsill, watching the world go by.

68. A pair of cats curled up together in a patch of sunlight.

69. A regal Abyssinian cat sitting on a velvet cushion, wearing a jeweled collar.

70. A fluffy Siberian cat exploring a snowy landscape.

71. A sleek black cat prowling through a forest at dusk.

72. A pair of Siamese kittens playing with a ball of yarn.

73. A ginger cat napping in a cozy bed.

74. A contented cat lounging in a sunbeam, surrounded by houseplants.

75. A playful kitten batting at a feather toy.

76. A regal British Shorthair cat sitting on a stack of books.

77. A fluffy Ragdoll cat being brushed by its owner.

78. A sleek Bengal cat stalking through tall grass.

79. A pair of cats snuggled up together in a cozy blanket.

80. A curious kitten peering out from inside a pumpkin.

81. A fluffy Maine Coon cat perched on a fence, watching birds.

82. A Siamese cat lounging in a sunbeam, surrounded by flowers.

83. A ginger cat playing with a ball of string.

84. A playful kitten chasing after a moth.

85. A majestic Persian cat sitting on a velvet cushion, adorned with jewels.

86. A black and white tuxedo cat sitting on a windowsill, framed by lace curtains.

87. A pair of cats curled up together in a hammock.

88. A regal Abyssinian cat sitting on a velvet throne, wearing a crown.

89. A fluffy Siberian cat exploring a snowy forest.

90. A sleek black cat prowling through a moonlit garden.

91. A pair of Siamese kittens cuddled up together on a fluffy rug.

92. A ginger cat napping in a cozy armchair.

93. A contented cat lounging in a sunbeam, surrounded by autumn leaves.

94. A playful kitten batting at falling snowflakes.

95. A regal British Shorthair cat sitting on a velvet cushion, wearing a bow tie.

96. A fluffy Ragdoll cat being cuddled by its owner, surrounded by candles.

97. A sleek Bengal cat stalking through a field of tall grass.

98. A pair of cats snuggled up together in a cozy bed.

99. A curious kitten peering out from inside a Christmas stocking.

100. A fluffy Maine Coon cat perched on a windowsill, watching the world go by.

Generated-cats Images

Generated-cats images as shown in Figure A.3

References

Xiao, B.; Wu, H.; Xu, W.; Dai, X.; Hu, H.; Lu, Y.; Zeng, M.; Liu, C.; and Yuan, L. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4818–4829.

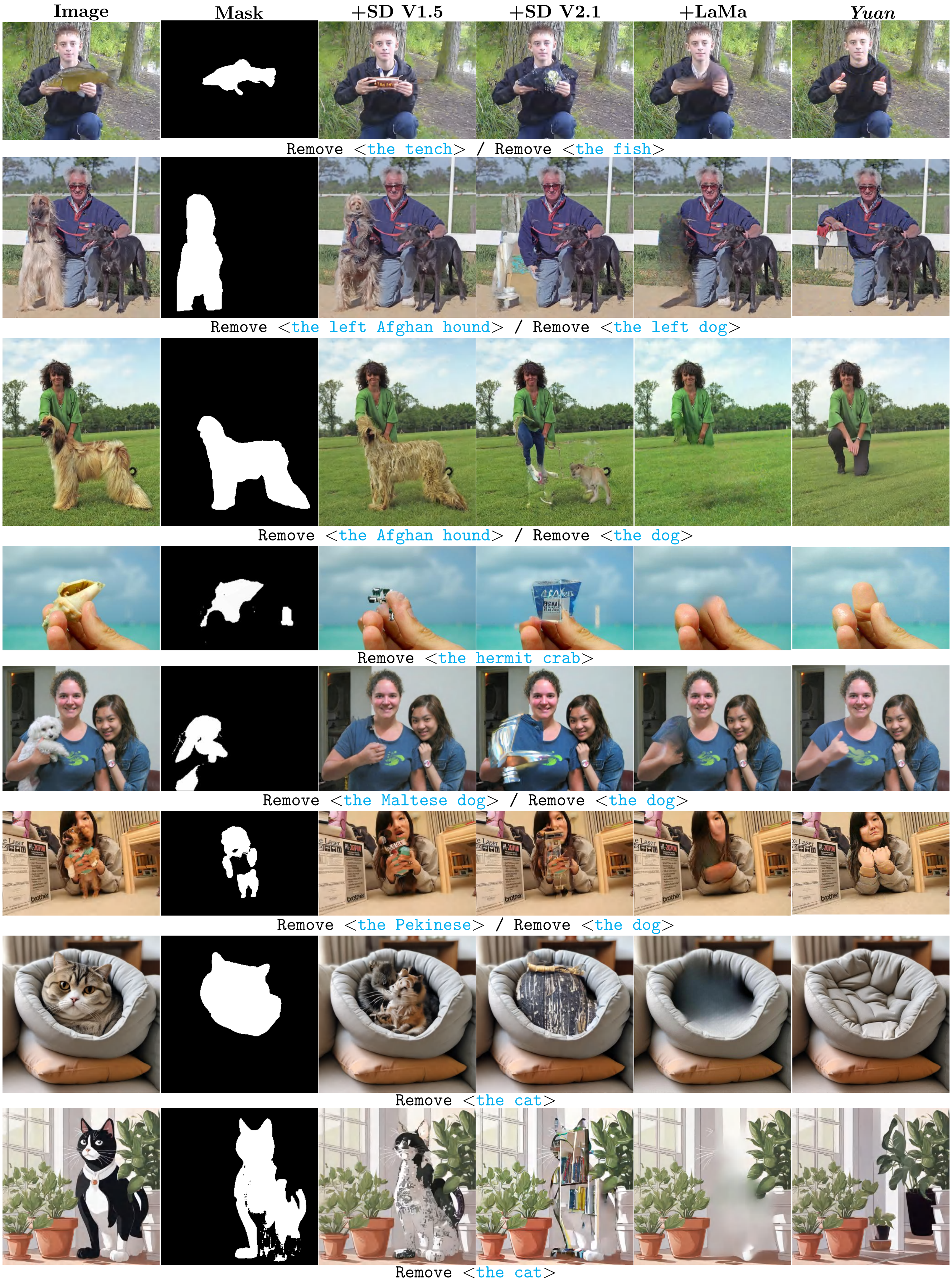
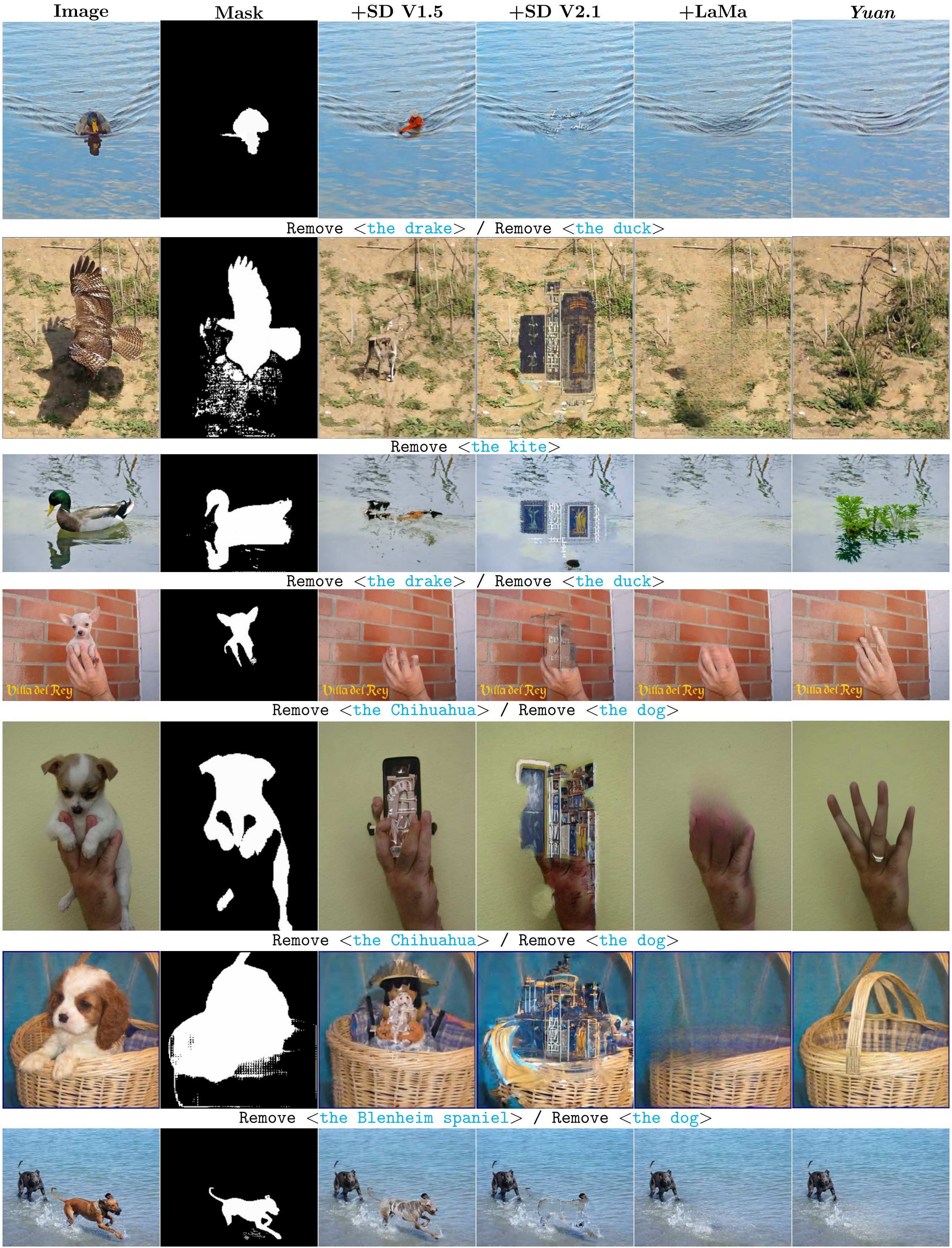


Image
Mask
+SD V1.5
+SD V2.1
+LaMa
Yuan
Remove <the tench> / Remove <the fish>
Remove <the left Afghan hound> / Remove <the left dog>
Remove <the Afghan hound> / Remove <the dog>
Remove <the hermit crab>
Remove <the Maltese dog> / Remove <the dog>
Remove <the Pekinese> / Remove <the dog>
Remove <the cat>
Remove <the cat>



Remove <the drake> / Remove <the duck>

Remove <the kite>

Remove <the drake> / Remove <the duck>

Remove <the Chihuahua> / Remove <the dog>

Remove <the Chihuahua> / Remove <the dog>

Remove <the Blenheim spaniel> / Remove <the dog>

Remove <the right Rhodesian ridgeback> / Remove <the right dog>

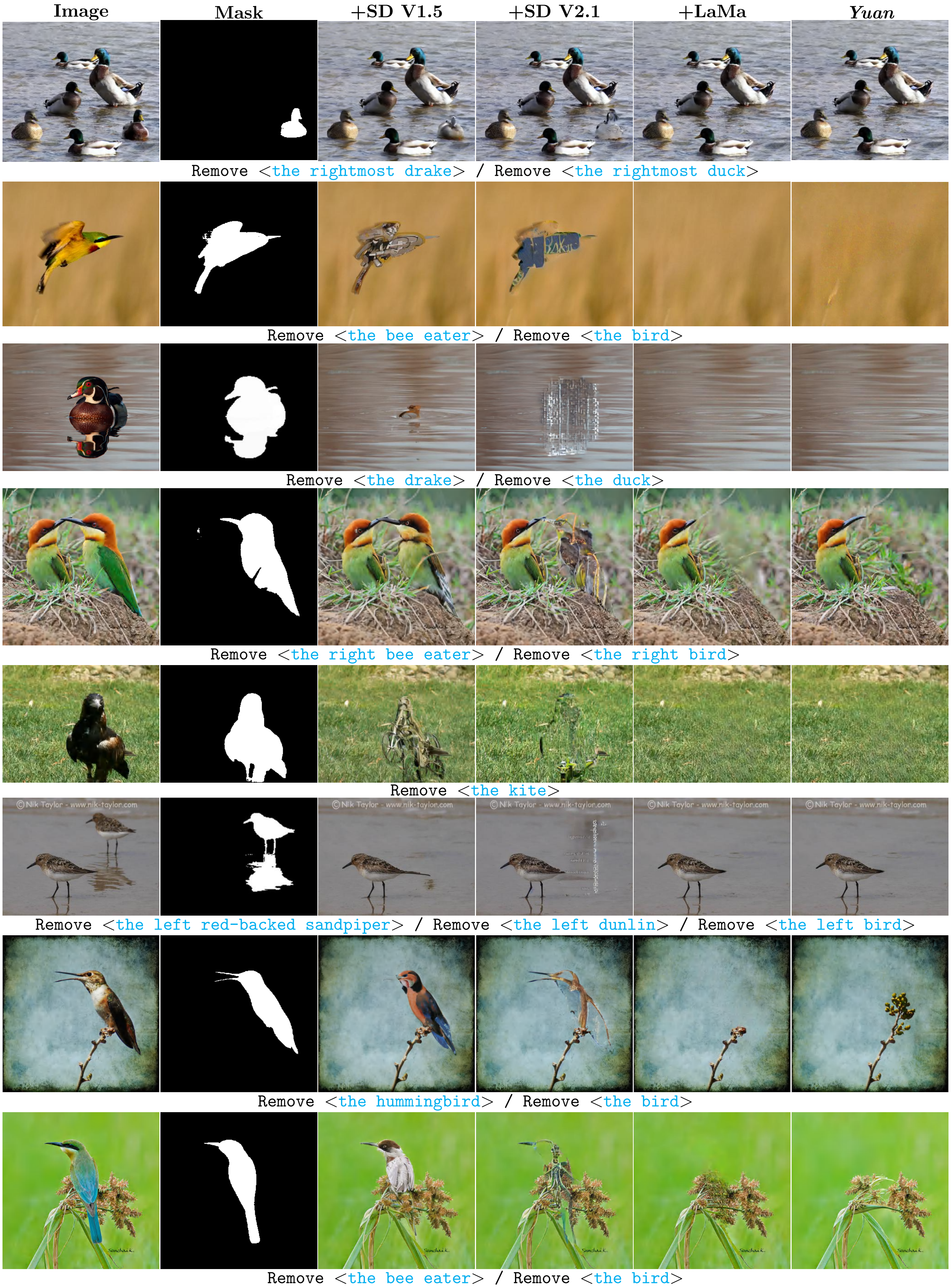


Image | Mask | +SD V1.5 | +SD V2.1 | +LaMa | *Yuan*

Remove <the rightmost drake> / Remove <the rightmost duck>

Remove <the bee eater> / Remove <the bird>

Remove <the drake> / Remove <the duck>

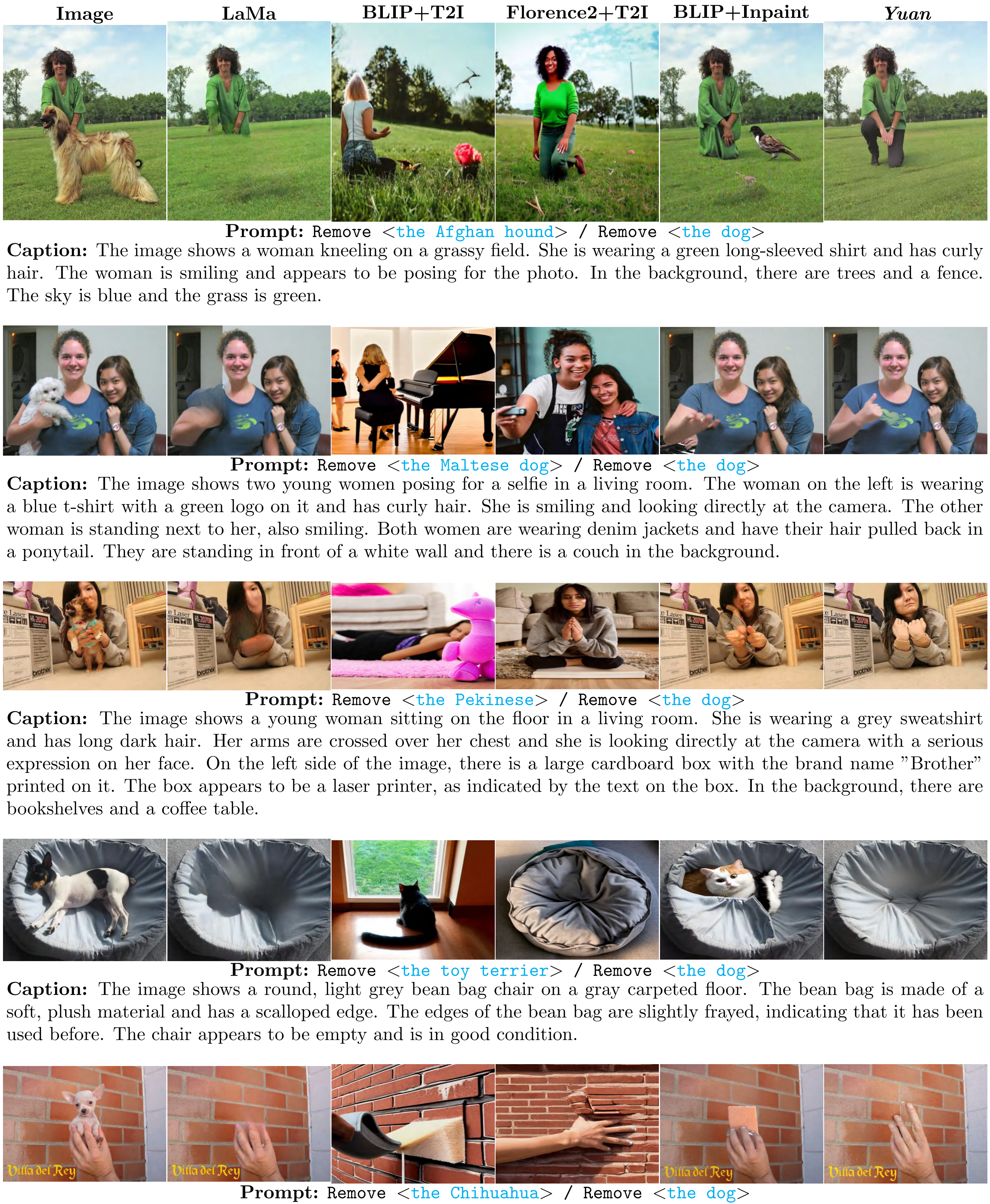
Remove <the right bee eater> / Remove <the right bird>

Remove <the kite>

Remove <the left red-backed sandpiper> / Remove <the left dunlin> / Remove <the left bird>

Remove <the hummingbird> / Remove <the bird>

Remove <the bee eater> / Remove <the bird>



Prompt: `Remove <the Afghan hound> / Remove <the dog>`

Caption: The image shows a woman kneeling on a grassy field. She is wearing a green long-sleeved shirt and has curly hair. The woman is smiling and appears to be posing for the photo. In the background, there are trees and a fence. The sky is blue and the grass is green.

Prompt: `Remove <the Maltese dog> / Remove <the dog>`

Caption: The image shows two young women posing for a selfie in a living room. The woman on the left is wearing a blue t-shirt with a green logo on it and has curly hair. She is smiling and looking directly at the camera. The other woman is standing next to her, also smiling. Both women are wearing denim jackets and have their hair pulled back in a ponytail. They are standing in front of a white wall and there is a couch in the background.

Prompt: `Remove <the Pekinese> / Remove <the dog>`

Caption: The image shows a young woman sitting on the floor in a living room. She is wearing a grey sweatshirt and has long dark hair. Her arms are crossed over her chest and she is looking directly at the camera with a serious expression on her face. On the left side of the image, there is a large cardboard box with the brand name "Brother" printed on it. The box appears to be a laser printer, as indicated by the text on the box. In the background, there are bookshelves and a coffee table.

Prompt: `Remove <the toy terrier> / Remove <the dog>`

Caption: The image shows a round, light grey bean bag chair on a gray carpeted floor. The bean bag is made of a soft, plush material and has a scalloped edge. The edges of the bean bag are slightly frayed, indicating that it has been used before. The chair appears to be empty and is in good condition.

Prompt: `Remove <the Chihuahua> / Remove <the dog>`

Caption: The image shows a hand of a person cleaning a brick wall. The wall is made up of red bricks arranged in a horizontal pattern. The bricks appear to be old and weathered, with some areas of the bricks having a rough texture. The hand is in the process of cleaning the wall, with the fingers slightly curled and the thumb and index finger pointing towards the right side of the image. The text "Villa del Rey" is written in yellow on the bottom right corner.

Figure A.2: The result of our works.

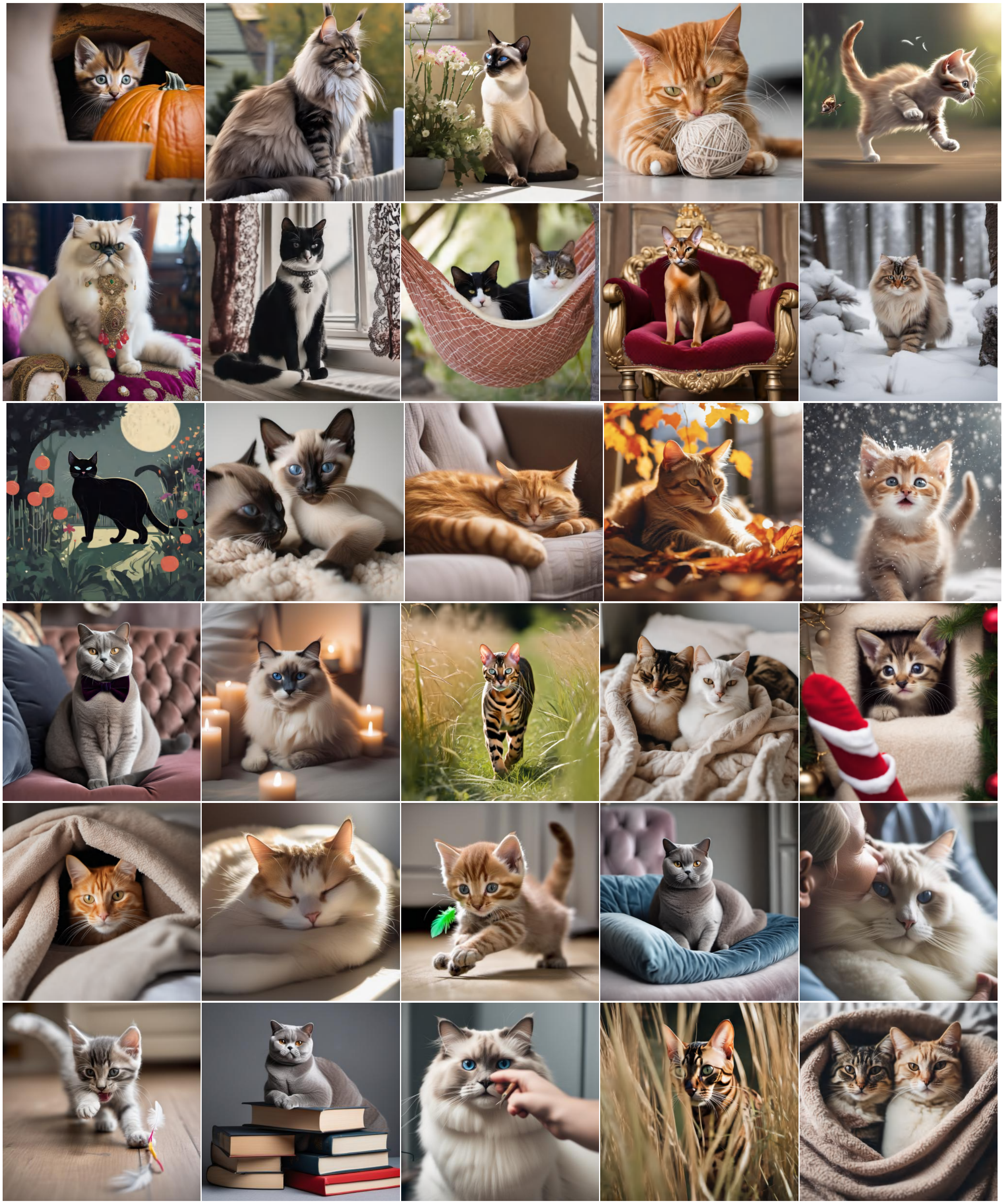


Figure A.3: Generated-cats images.