

---

# Supplemental Material to “Benchmark for Compositional Text-to-Image Synthesis”

---

## 1 Appendix

In the following we provide additional training details (Section 1.1), details on our human evaluation (Section 1.2), additional qualitative examples (Section 1.3) and in-depth statistics for our new compositional splits (Section 1.4).

### 1.1 Training Details

The models (DMGAN, ControlGAN, DFGAN) that we evaluated on the proposed benchmark follow the default hyperparameters used in their respective codebases: <https://github.com/MinfengZhu/DMGAN>, <https://github.com/mr1ibw/ControlGAN>, <https://github.com/tobran/DF-GAN>. With regards to CLIP-R-Precision, we take the pretrained RN101 model from <https://github.com/openai/CLIP> and finetune the model on the CUB and Oxford-Flowers datasets. Since CUB and Oxford-Flowers datasets are relatively small in size compared to the dataset that CLIP was pretrained on, we only finetune the final few layers. The exact layers we finetune are [“visual.layer4”, “visual.attnpool”, “transformer.resblocks.11”, “ln\_final”, “text\_projection”, “logit\_scale”]. We finetune the model using AdamW optimizer with a learning rate of 0.0005 and batch size of 256 for 100 epochs using the same contrastive loss used for pretraining the original CLIP model. The finetuned CLIP models and the code for computing the CLIP-R-Precision will be publically available. The experiments in this paper were run using NVIDIA Tesla P100 16G GPUs and Titan Xp 12G GPUs.

### 1.2 Human Evaluation on AMT

We conducted human evaluation on Amazon Mechanical Turk (AMT). We had two types of tasks: perceptual evaluation and correctness evaluation. Figure 1 shows the interface for the perceptual evaluation (identical for both C-CUB and C-Flowers datasets). Figures 2 and 3 show the interfaces for the correctness evaluation for C-CUB and C-Flowers, respectively (mostly identical except for a few details). We paid \$0.1 for each perceptual task and \$0.2 for each correctness task, \$2,400 in total, on average estimated \$5 per hour. One caveat is that time spent on the task varied significantly by worker: some were extremely fast, *e.g.* 20 seconds on average, while others took much longer, *e.g.* 100 seconds on average, with no visible gap in quality.

Instructions

Given two images, please, judge which one looks more realistic.

- Please, try to make a judgment on whether Image 1 or Image 2 looks more **realistic**.
- If unable to decide, select "About the same", but please, **use this option sparingly!**
- Note, that some images may be slightly truncated but that alone does not mean they are not realistic, i.e., we are mainly interested in **perceptual quality** (which image looks "nicer").



We anticipate that it takes at **the very least** 5-10 seconds to perform a task (in "easy" cases). If a worker **consistently** submits answers after 3-4 seconds or so, we will reject such submissions and consider blocking the worker.

We also check the results and reject all tasks from workers, who obviously did not follow the instructions.

Thank you for reading and cooperation!

Image 1:

Image 2:

Which image looks more realistic?

✓ - select one -

1 - Image 1  
2 - Image 2  
3 - About the same (USE SPARINGLY)

Figure 1: AMT Interface for the perceptual quality evaluation (the same for C-CUB and C-Flowers).


**Instructions**

Given an image and two captions, judge which caption better matches the image.

- Please, try to make a judgment on whether Caption 1 or Caption 2 **better** matches the image (even if neither is perfect).
- If unable to decide, select "About the same", but please, **use this option sparingly!**
- In this task the images/captions are about **birds**. If you encounter some terms that you do not know, please, consult the diagrams linked below.

Please **read** the captions carefully! We anticipate that it takes at **the very least** about 10 seconds to perform a task (in "easy" cases). If a worker **consistently** submits answers after 3-4 seconds or so, we will reject such submissions and consider blocking the worker. We also check the results and reject all tasks from workers, who obviously did not follow the instructions. Thank you for reading and cooperation!

**Image:**



Caption 1: this bird has blue neck, throat, crown and black wings, belly and breast, tail.

Caption 2: this bird has yellow neck,throat, crown and black wings, belly and breast, tail.

**Which caption better matches the image?**

✓ - select one -

1 - Caption 1

2 - Caption 2

3 - About the same (USE SPARINGLY)

Figure 2: AMT Interface for the correctness evaluation on C-CUB.

Instructions

Given an image and two captions, judge which caption better matches the image.

- Please, try to make a judgment on whether Caption 1 or Caption 2 **better** matches the image (even if neither is perfect).
- If unable to decide (or e.g. image quality is too bad), select "About the same", but please, **use this option sparingly!**
- In this task the images/captions are about **flowers**. If you encounter some terms that you do not know, please, consult the diagrams linked below.

Please **read** the captions carefully! We anticipate that it takes at **the very least** about 10 seconds to perform a task (in "easy" cases). If a worker **consistently** submits answers after 3-4 seconds or so, we will reject such submissions and consider blocking the worker. We also check the results and reject all tasks from workers, who obviously did not follow the instructions. Thank you for reading and cooperation!

Image:

Caption 1: the petals of this flower are magenta with a oval stigma

Caption 2: the petals of this flower are magenta with a long stigma

Which caption better matches the image?

✓ - select one -

1 - Caption 1  
2 - Caption 2  
3 - About the same (USE SPARINGLY)

Figure 3: AMT Interface for the correctness evaluation on C-Flowers.

### 28 1.3 Qualitative Examples

29 We present additional qualitative examples from DMGAN on all four of our Seen/Swapped splits in  
 30 Figures 4, 5, 6, 7.

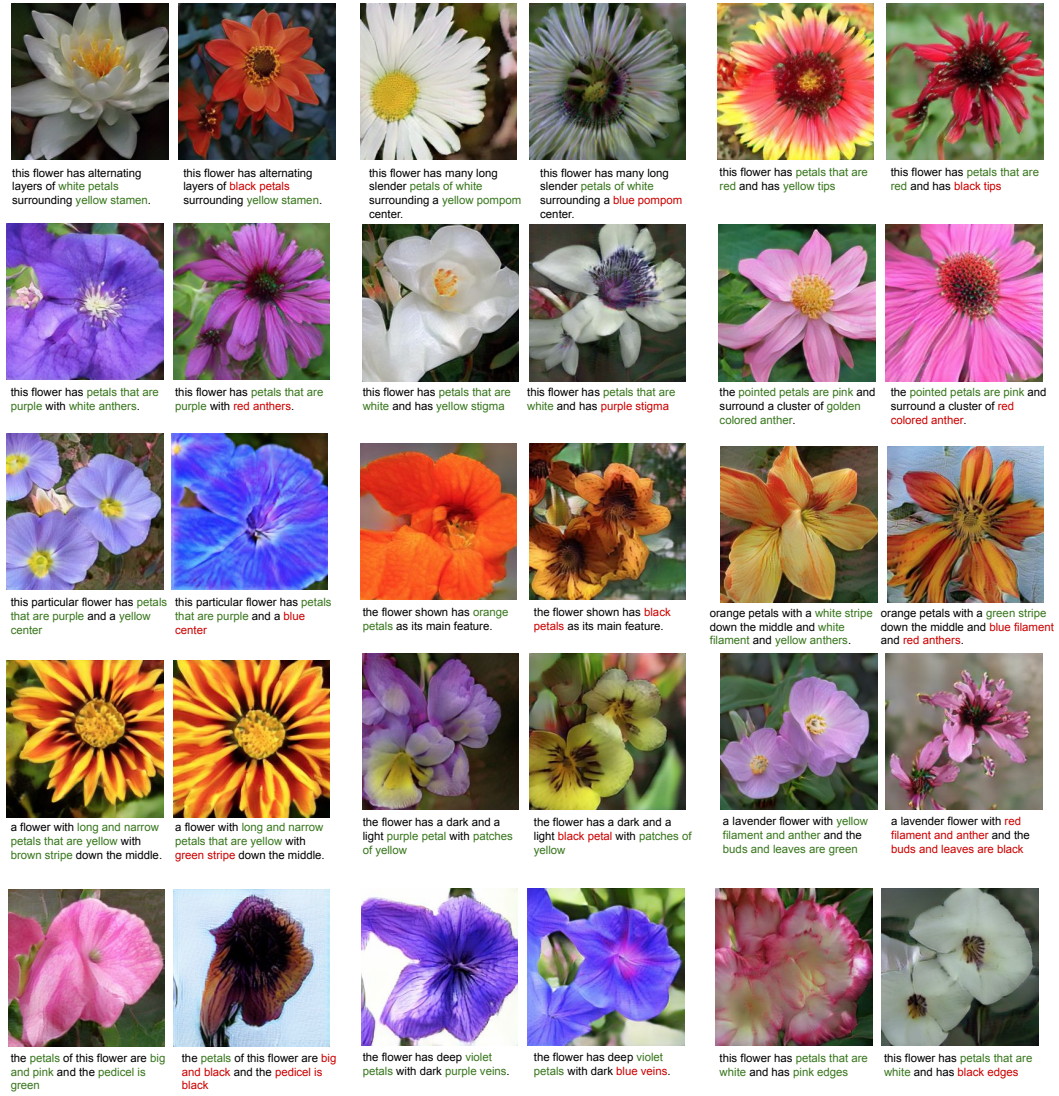


Figure 4: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-Flowers color dataset.



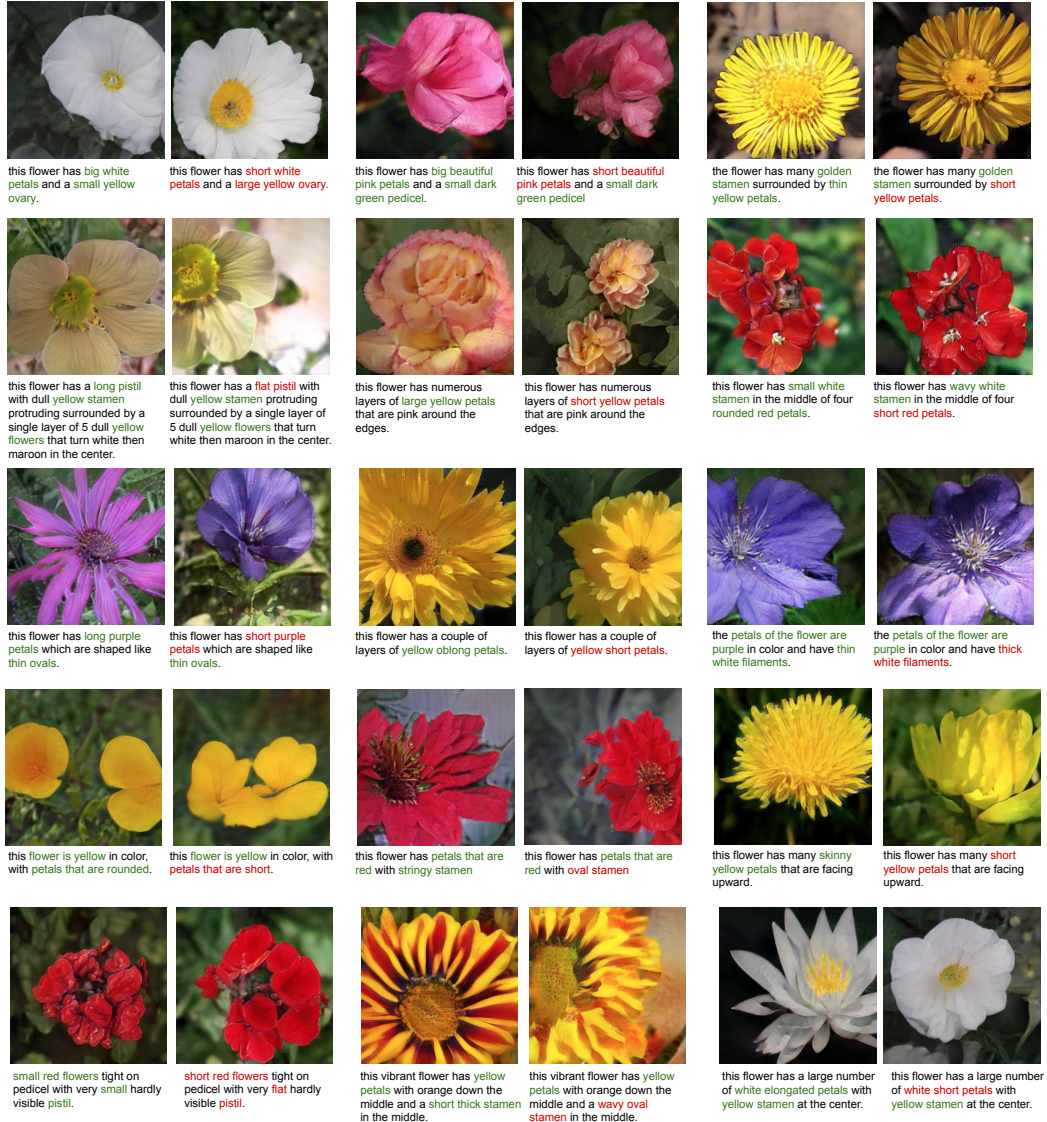


Figure 5: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-Flowers shape dataset.



Figure 6: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-CUB color dataset.





Figure 7: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-CUB shape dataset.



## 31 1.4 Compositional Splits Statistics

32 In this section, we present additional statistics for the C-CUB and C-Flowers datasets. With regards  
33 to C-CUB color, by comparing the training split in Figure 8 and test Unseen split in Figure 10, we  
34 see that the overall adjective and noun distributions are similar between the two even though the test  
35 Unseen split contains novel compositions. However, comparing the training split with test Swapped  
36 split in Figure 11, we notice that the adjective and noun distributions shift quite significantly due to  
37 the swaps. For instance, adjectives “purple” and “tan” become significantly more frequent, resulting  
38 in pairs like “purple bill” and “tan wing” to dominate the split. Similar observations can be made in  
39 other compositional splits (*e.g.* C-CUB Shape, C-Flowers Color, C-Flowers Shape). Thus, it becomes  
40 important to account for these imbalances during evaluation for test swapped splits.

## 41 1.5 Candidate Balancing for Evaluation on Test Swapped

42 As mentioned in the previous section, additional balancing needs to be done in order to prevent  
43 certain heldout pairs from overwhelming the metric computation. To do so, we first gather all the  
44 captions and the generated images by concept pairs. Then, the most frequent pair(s) is identified and  
45 the caption-image pairs belonging to that dominant pair(s) are subsampled so that their numbers  
46 are only 1.25x larger than the next most dominant pair. The code block for such candidate balancing  
47 is provided in the following:

```

48 1 """
49 2 gather_by_pair:
50 3 Dictionary that maps adj_noun pair to a list of entries that contain
51   that pair.
52 4 An entry contains (conditioned text, generated_img)
53 5 dtype: "bird" or "flower"
54 6 split: "color" or "shape"
55 7 """
56 8 def balance_candidates(gather_by_pair, dtype, split):
57 9     change_type_counts = Counter()
58 10    for change_type, data in gather_by_pair.items():
59 11        change_type_counts[change_type] = len(data)
60 12    top3 = change_type_counts.most_common(3)
61 13
62 14    """
63 15    for CUB color split where "purple bill" and "tan wing" are
64 16    dominant
65 17    """
66 18    if dtype == 'bird' and split == 'color':
67 19        max_num_dominant = int(min(top3[0][1], 1.25 * top3[-1][1]))
68 20        dominant_1 = top3[0][0]
69 21        dominant_2 = top3[1][0]
70 22
71 23        dominant_1_cands = \
72 24            random.sample(gather_by_pair[dominant_1],
73 25                           max_num_dominant)
74 26        dominant_2_cands = \
75 27            random.sample(gather_by_pair[dominant_2],
76 28                           max_num_dominant)
77 28
78 29        all_cands = []
79 30        for pair, entries in gather_by_pair.items():
80 31            if pair == dominant_1 or pair == dominant_2:
81 32                continue
82 33            all_cands += entries
83 34        all_cands += dominant_1_cands + dominant_2_cands
84 35    else:
85 36        max_num_dominant = int(min(top3[0][1], 1.25 * top3[1][1]))
86 37        dominant = top3[0][0]
87 38
88 39        dominant_cands = random.sample(gather_by_pair[dominant],
89 40                                       max_num_dominant)
90 41        all_cands = []
91 42        for pair, entries in gather_by_pair.items():
92 43            if pair == dominant:
93 44                continue
94 45            all_cands += entries
95 46        all_cands += dominant_cands
96 47
97 48    return all_cands

```

Listing 1: Candidate Balancing for Evaluation on Test Swapped

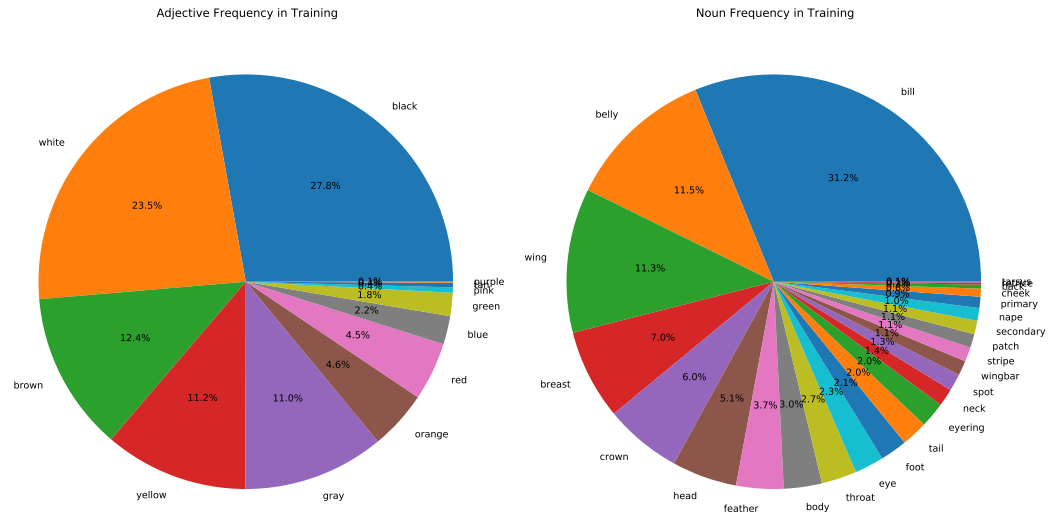


Figure 8: Adjective and noun distributions of training set for C-CUB Color.

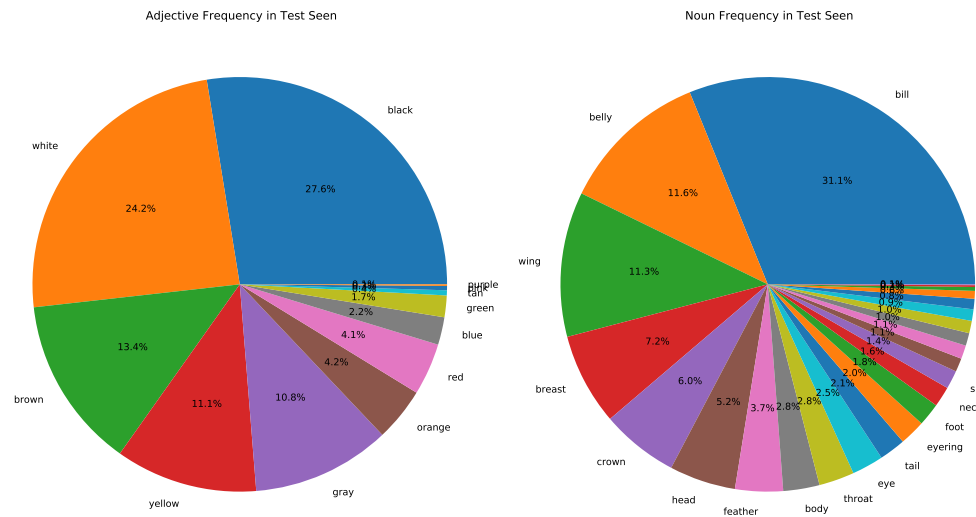


Figure 9: Adjective and noun distributions of test SEEN set for C-CUB Color.

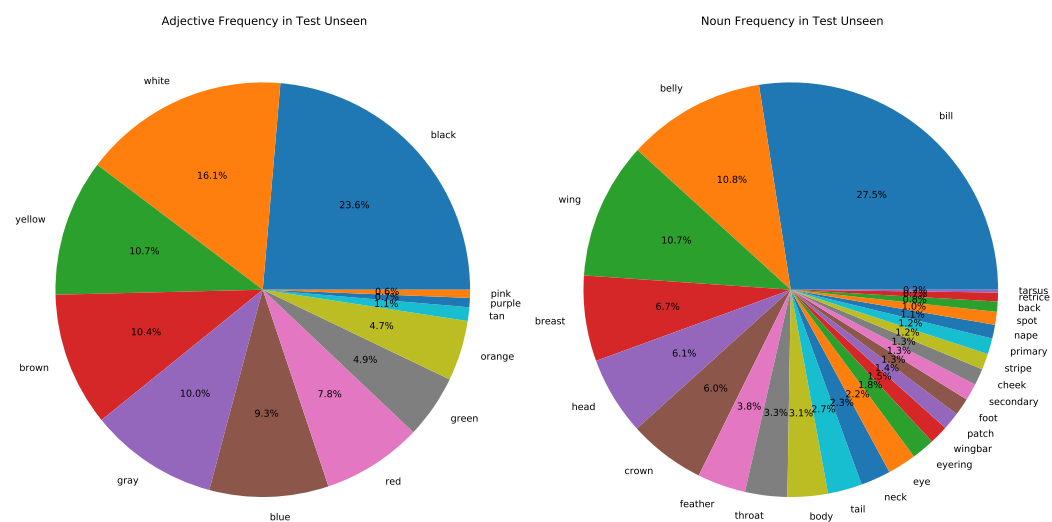


Figure 10: Adjective and noun distributions of test UNSEEN set for C-CUB Color.

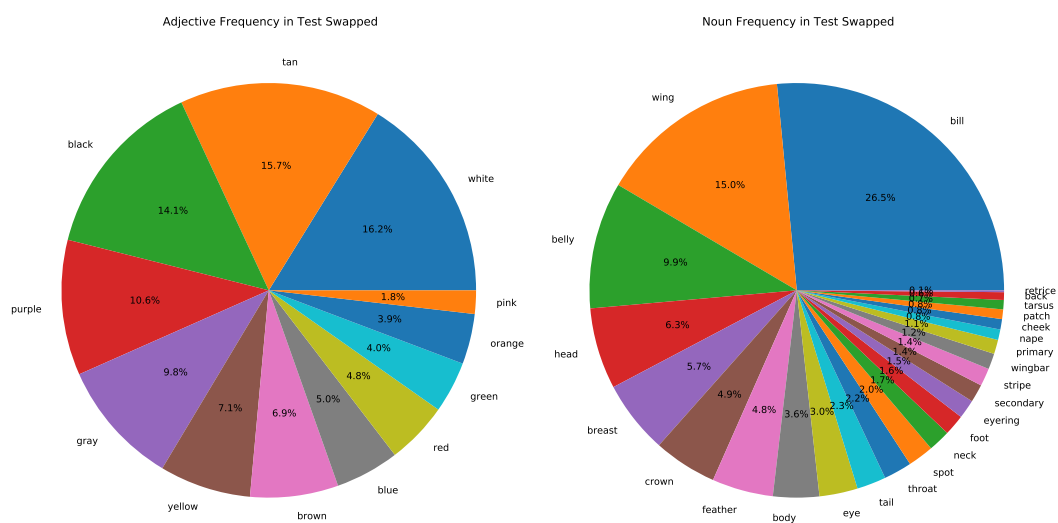


Figure 11: Adjective and noun distributions of test SWAPPED set for C-CUB Color.



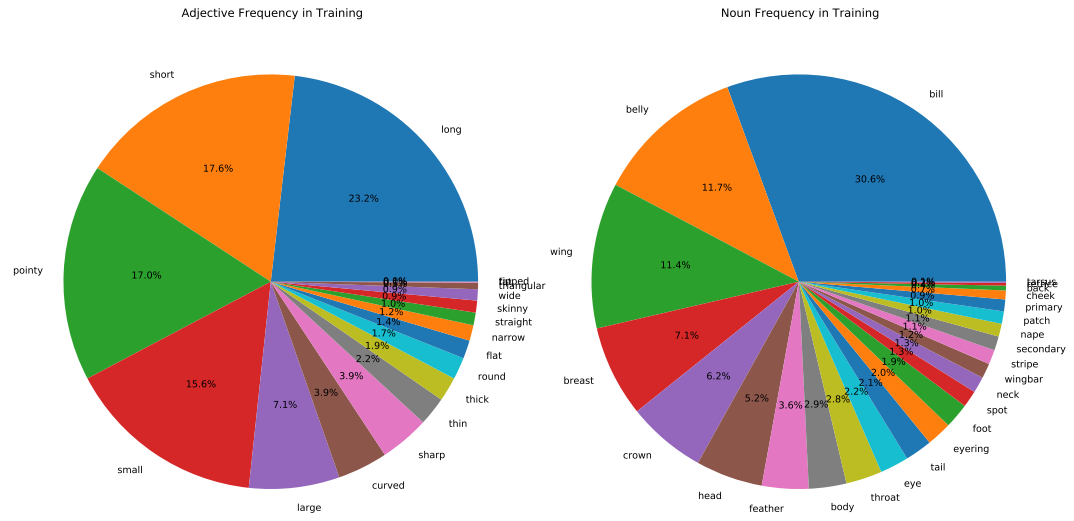


Figure 12: Adjective and noun distributions of training set for C-CUB Shape.

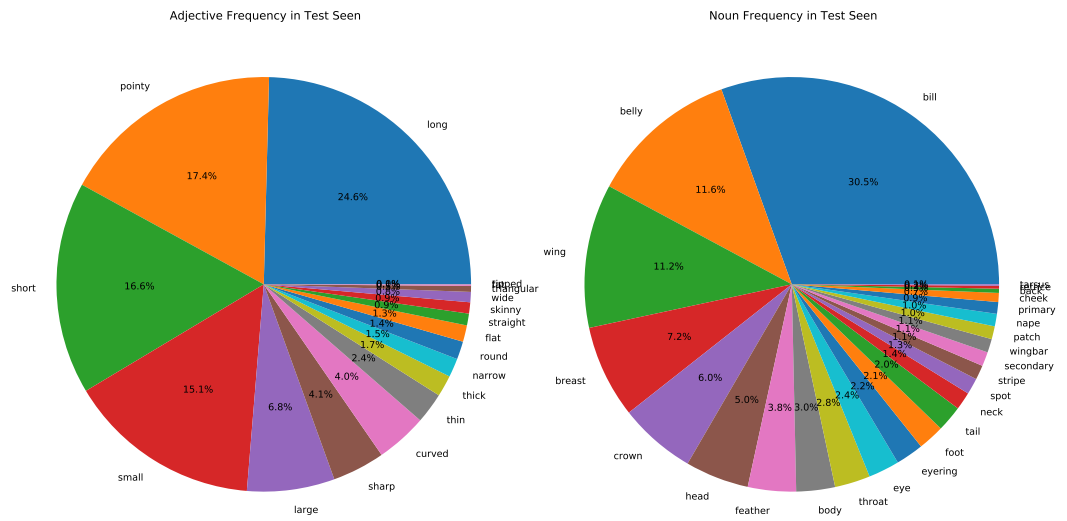


Figure 13: Adjective and noun distributions of test SEEN set for C-CUB Shape.

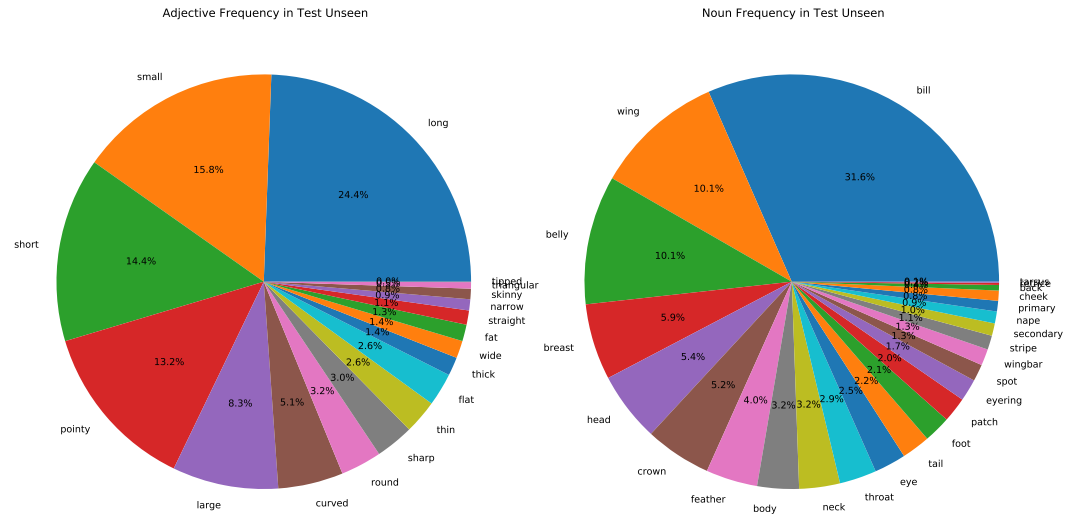


Figure 14: Adjective and noun distributions of test UNSEEN set for C-CUB Shape.

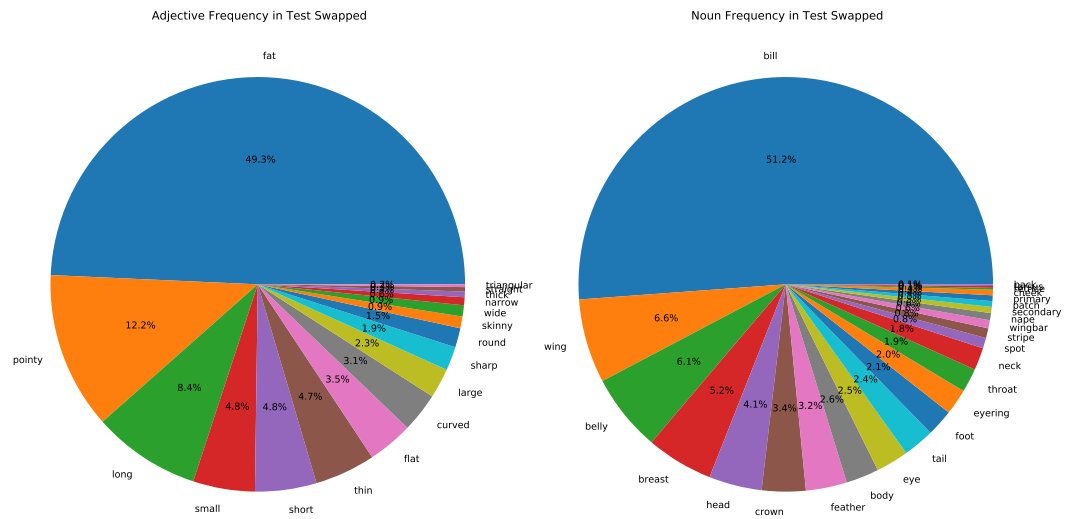


Figure 15: Adjective and noun distributions of test SWAPPED set for C-CUB Shape.

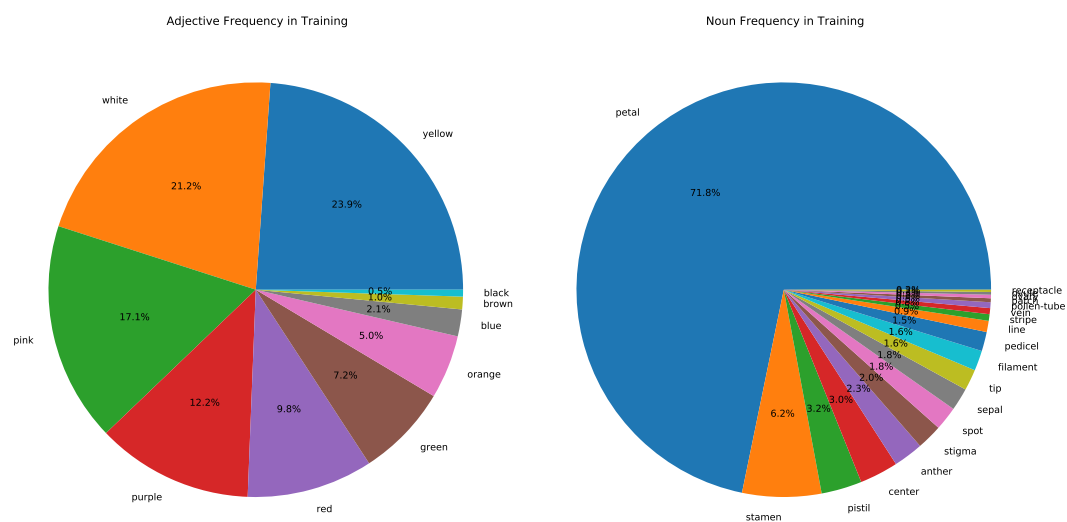


Figure 16: Adjective and noun distributions of training set for C-Flowers Color.

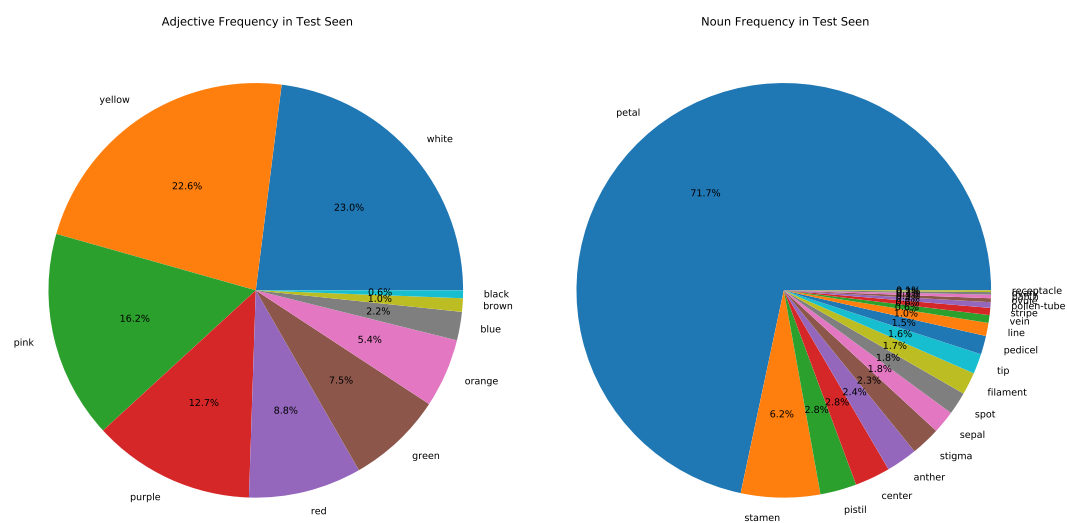


Figure 17: Adjective and noun distributions of test SEEN set for C-Flowers Color.

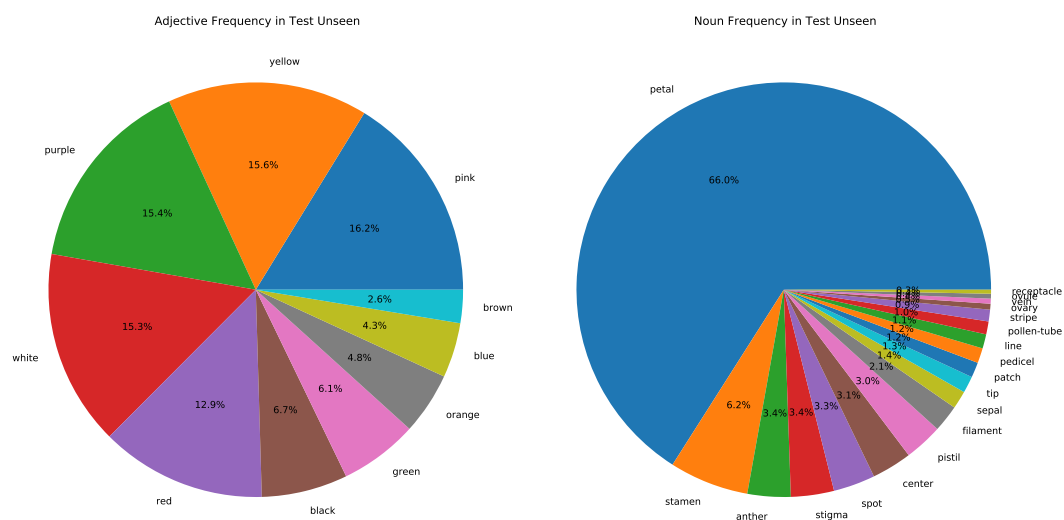


Figure 18: Adjective and noun distributions of test UNSEEN set for C-Flowers Color.

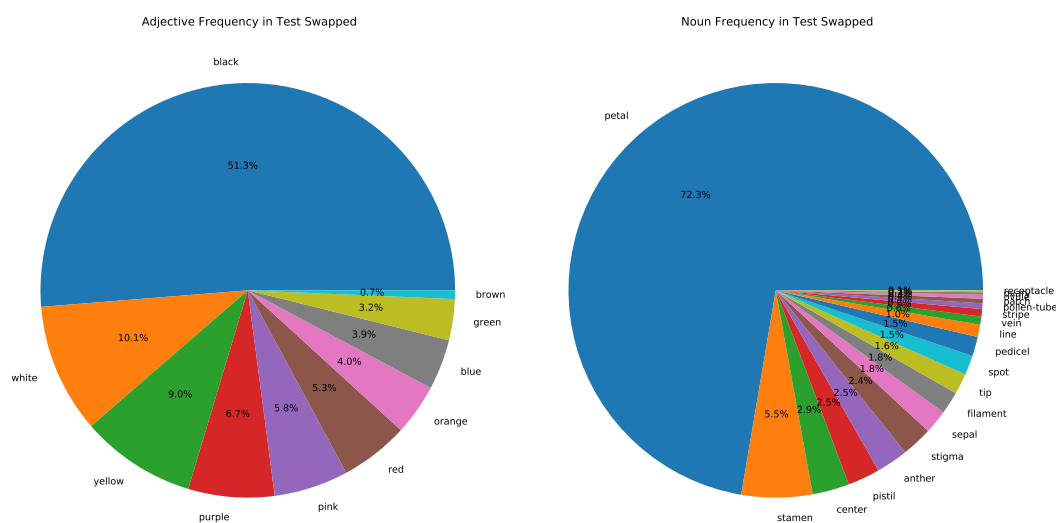


Figure 19: Adjective and noun distributions of test SWAPPED set for C-Flowers Color.



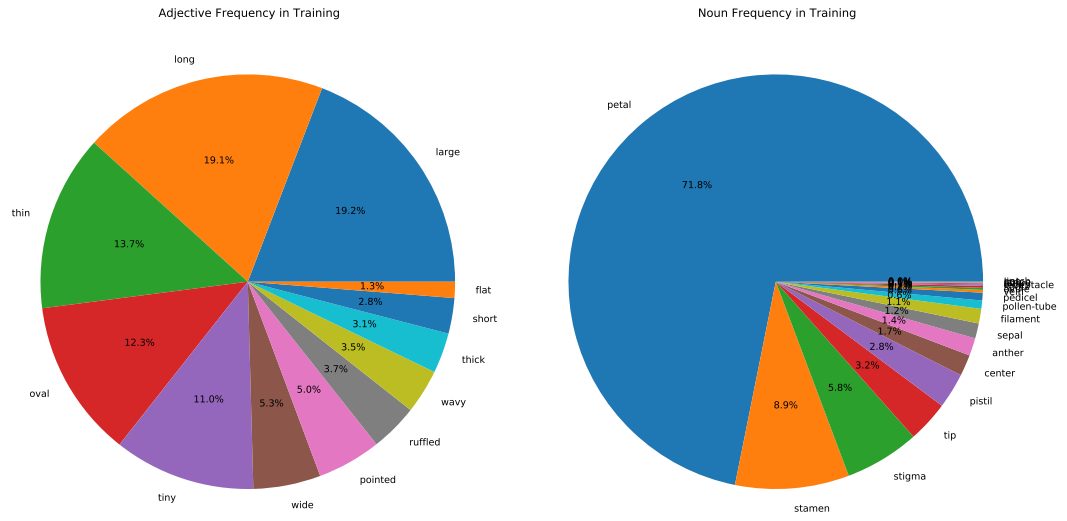


Figure 20: Adjective and noun distributions of training set for C-Flowers Shape.

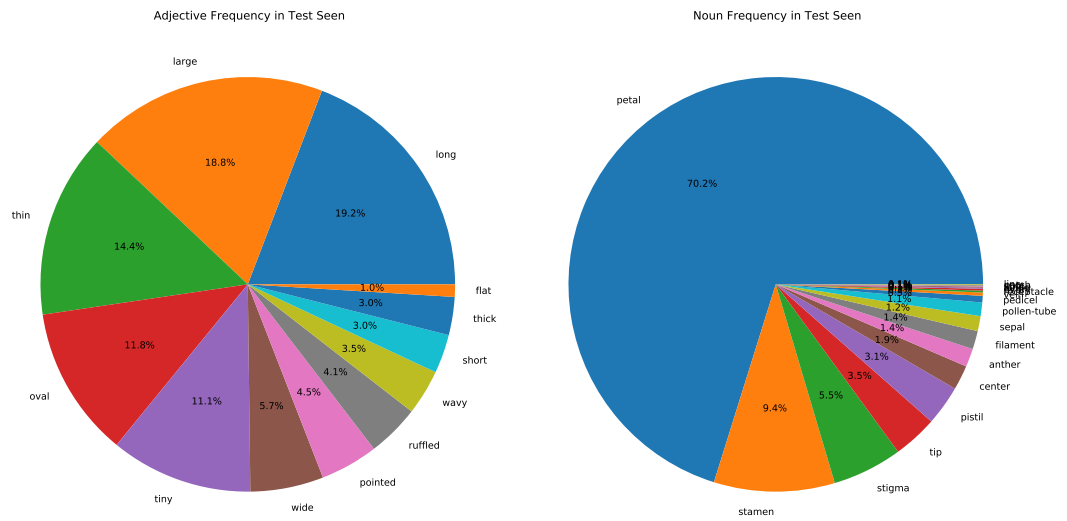


Figure 21: Adjective and noun distributions of test SEEN set for C-Flowers Shape.

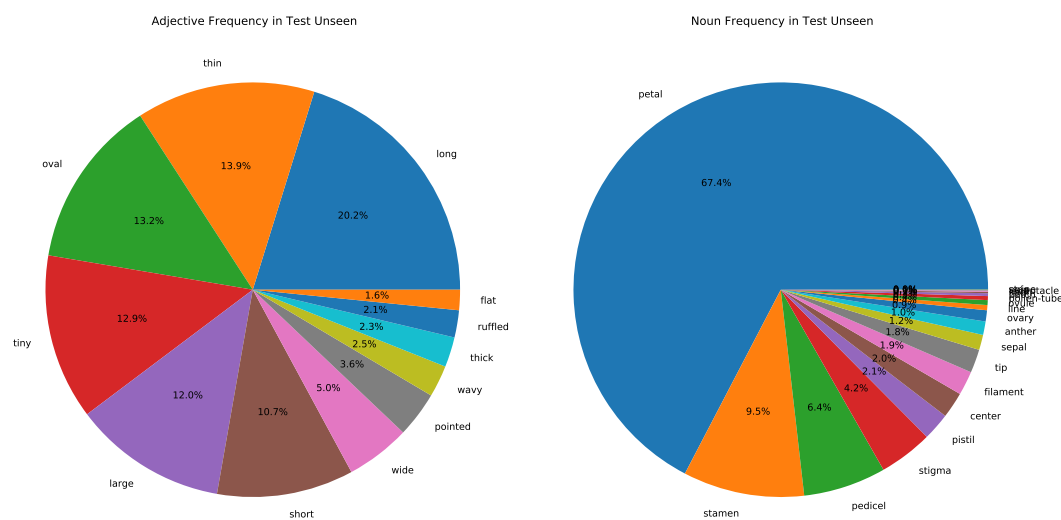


Figure 22: Adjective and noun distributions of test UNSEEN set for C-Flowers Shape.

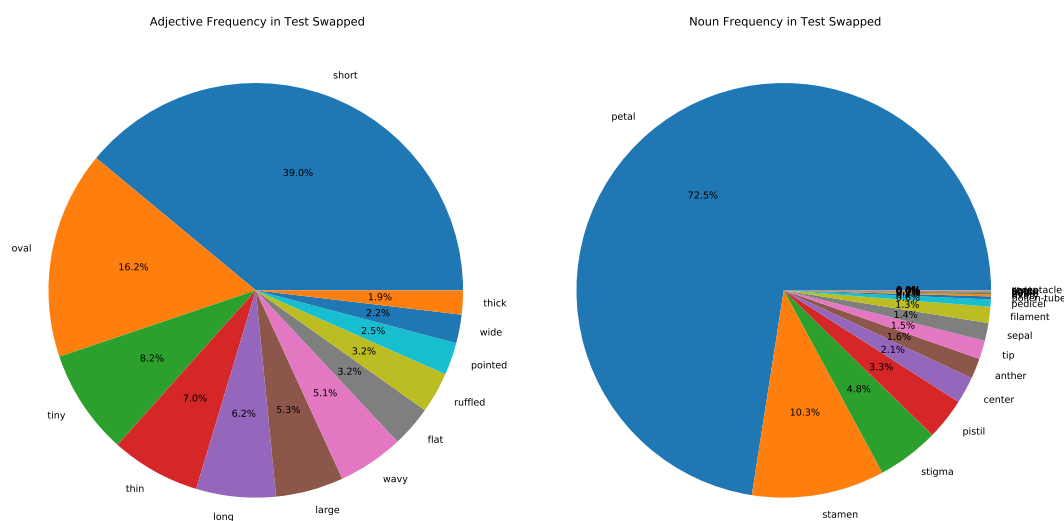


Figure 23: Adjective and noun distributions of test SWAPPED set for C-Flowers Shape.