Appendix: Zero-guidance Segmentation Using Zero Segment Labels

In this Appendix, we provide additional details and experiments:

- Section A: CLIP's self-attention visualization
- Section B: Implementation details of our segment candidate finding method
- Section C: Thresholds used in metrics
- Section D: Limitations of Pascal VOC dataset for evaluation.
- Section E: Details on hyperparameters tuning
- Section F: User study
- Section G: Additional results
- Section H: Potential negative societal impacts

A. CLIP's Self-attention Visualization

Figure 10 visualizes the self-attention maps of CLIP's image encoder across different layers. The self-attention maps appear to be meaningful in the earlier layers, i.e., the patch tokens mostly attend to regions that contain semantically similar pixels, and the global token attends to regions with prominent objects. However, the self-attention map appear more random and uninterpretable in the later layers.

B. Finding Segment Candidates with DINO: Implementation Details

We provide more implementation details for Section 3.1. We adopt DINO feature extraction method from Amir et al. [1]. The method first feeds an input image into DINO and extracts "key" values from the last attention layer as dense spatial features.

After extracting the features, we partition the image into segments by clustering DINO's features. We perform bottom-up clustering starting from each feature vector. The merging is done recursively by combining two clusters with the least combined variance. After this initial clustering, we end up with a binary tree where the root is the cluster of all the feature vectors. This binary tree structure is used as a heuristic to perform divisive clustering. Each node in the tree is represented by the average feature of its members. We prune the siblings whose cosine similarity score is over $T_{Dino} = 0.9$. This yields a segmentation map with all leaf nodes of the binary tree as segments. The two-stage clustering algorithm is chosen to lessen the computation requirement since we start from a large number of spatial features (111×111).

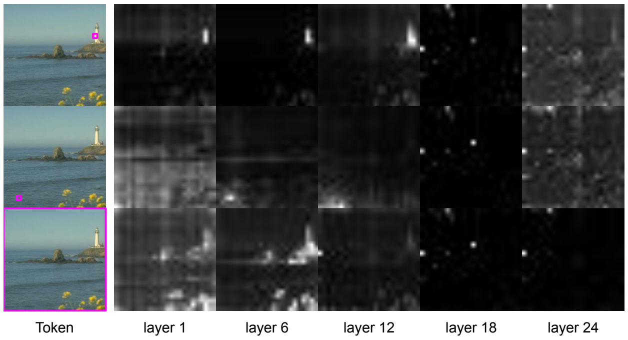


Figure 10. **Visualization of self-attention in CLIP's image encoder.** Each row shows the attention of the token of the pink patch across layers. The last row shows global token's attention.

Following Amir et al, the segmentation map is then upsampled to input resolution and refined using DenseCRF as described in [18]. The Unary Energy is set as the normalized distance of each feature vector to all k centroids, and the pairwise connection is fully-connected. Pairwise edge potentials are Gaussian kernels with location (pixel coordinates) as feature and Bilateral kernels with location and RGB values as features. Our implementation can be founded in the provided source code.

C. Thresholds Used in Metrics

S-BERT text-to-text similarity threshold (τ_{SBERT}). We provide Text-to-text IoU (IoU_{tt}) scores with several τ_{SBERT} threshold values in Figure 11 and Table 6. In the main experiment, when referred to a constant threshold, we select $\tau_{\text{SBERT}} = 0.5$ as it represents an approximate minimum threshold that human evaluators use to determine if two sentences share a common topic, based on a user study [3, 10].

CLIP segment-to-text similarity threshold (τ_{CLIP}). We provide Segment-to-text IoU (IoU_{st}) scores with several τ_{CLIP} threshold values in Figure 12 and Table 7. Selecting the threshold τ_{CLIP} is more challenging, since there is no established consensus or user studies to rely on. Figure 13 shows histograms of CLIP similarity scores between ground-truth image segments and their corresponding ground-truth labels in Pascal Context and Pascal VOC datasets. Given the distributions, we select $\tau_{\text{CLIP}} = 0.1$ to be on the safe side to report Segment-to-text IoU scores in the main experiment.

It is important to note that for our zero-guidance segmentation problem, the thresholds τ_{CLIP} and τ_{SBERT} are used in the label reassignment verification process (Section 4.2), which is part of the evaluation not the segmentation algorithm itself. For a given algorithm, varying the threshold values can result in distinct performance profiles, e.g., a precision-recall curve, and several thresholds may be used together for the purposes of evaluation and comparison, as is common practice in the object detection literature [41].

IoU threshold (τ_{IoU}). We use $\tau_{\text{IoU}} = 0.5$, which is com-

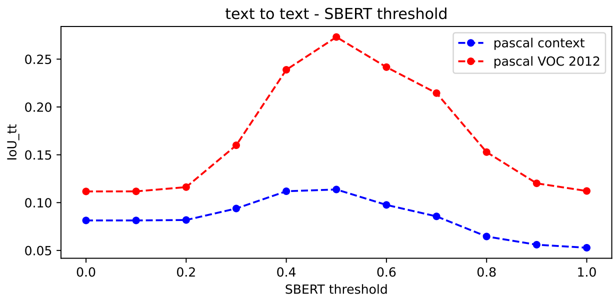


Figure 11. **Text to text IoU and SBERT threshold**

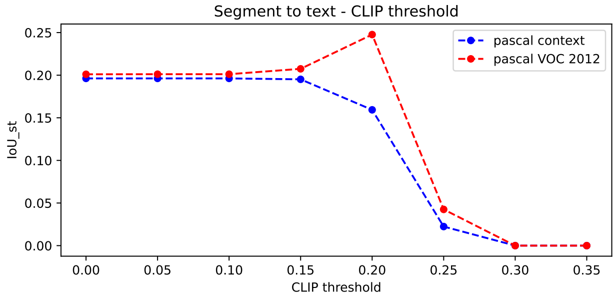


Figure 12. **Segment to text IoU and CLIP threshold**

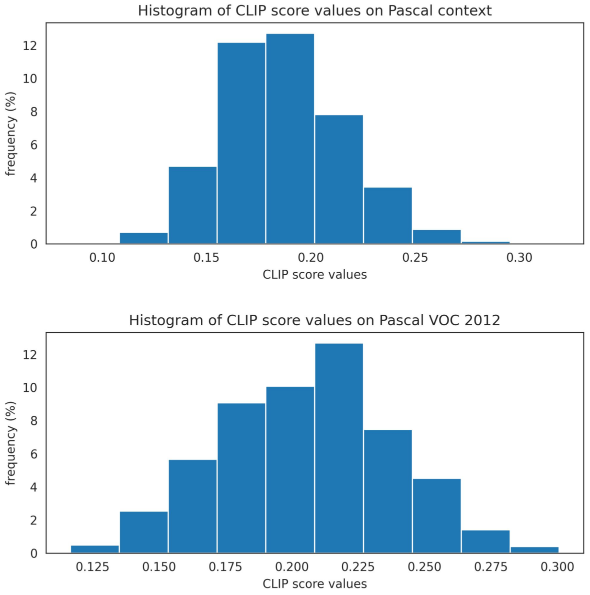


Figure 13. **CLIP similarity score distribution between the ground truth segment and the ground truth label**

monly used in object detection tasks to determine if a predicted bounding box is 'correct' compared to the ground truth [41].

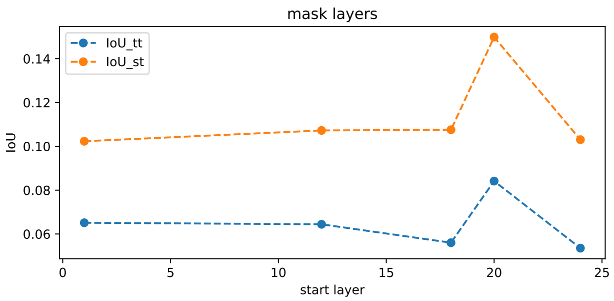


Figure 14. **Start masking layer selection**

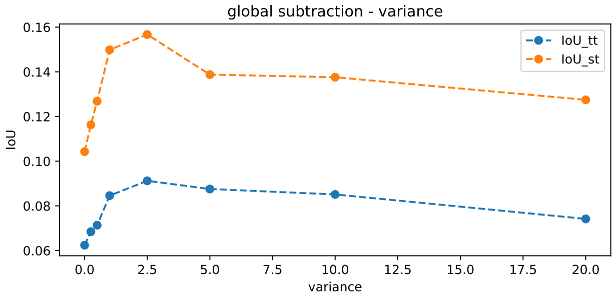


Figure 15. **Global subtraction's variance selection**

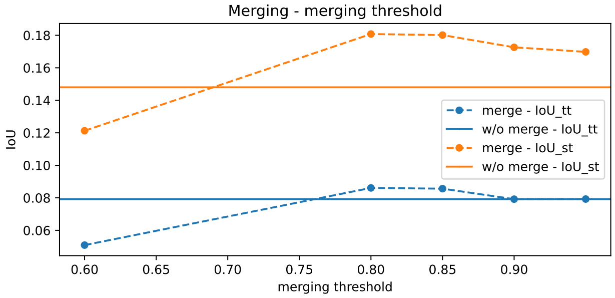


Figure 16. **Merging threshold selection**

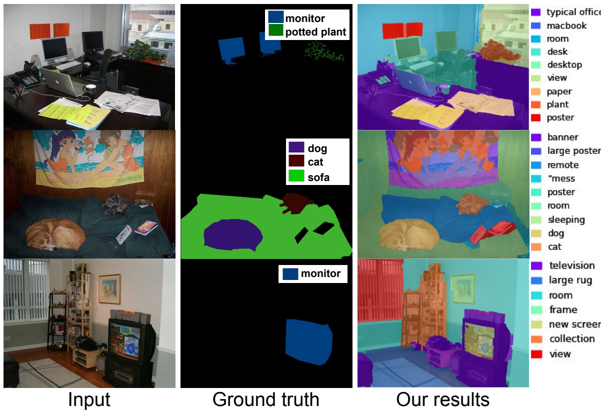


Figure 17. An example of Pascal VOC segmentation dataset

Table 6. Text-to-text IoU with several SBERT thresholds

					IoU$_{\text{tt}}$						
τ_{SBERT}	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Pascal Context	8.1	8.1	8.2	9.4	11.2	11.4	9.8	8.6	6.5	5.6	5.3
Pascal VOC	11.2	11.2	11.6	16.0	23.9	27.3	24.2	21.5	15.3	12.0	11.2

Table 7. Segment-to-text IoU with several CLIP thresholds

					IoU$_{\text{st}}$			
τ_{CLIP}	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
Pascal Context	19.6	19.6	19.6	19.5	16.0	2.2	0.0	0.0
Pascal VOC	20.1	20.1	20.1	20.8	24.8	4.2	0.0	0.0

D. Limitations of Pascal VOC for evaluating zero-guidance segmentation.

Evaluating zero-guidance segmentation performance using Pascal VOC (PAS-20) may not be ideal because PAS-20 has a very small number of labeled classes. In this dataset, many objects or sometimes the vast majority of regions in the images are left unlabeled as shown in Figure 17. Our method can discover various objects not presented in the ground truth labels, such as ‘paper’, ‘macbook’, and ‘poster’, but these are never counted towards any IoU scores.

E. Hyperparameters Tuning

We present how our hyperparameters, which are the layer to start attention-masking, the global subtraction variance, and the merging threshold, are tuned. Our tuning metrics are the Text-to-text IoU (IoU$_{\text{tt}}$) and Segment-to-text IoU (IoU$_{\text{st}}$) with the constant thresholds. The data used in this process are 100 randomly selected images from the Pascal Context's training split, which is never used for evaluation. Note here that there is no training involved in our pipeline.

The first parameter is the layer where attention masking starts. We found that masking from early layers erases all global context, resulting in poor results as context can be crucial for recognizing objects. Masking only the last layer also has poor results due to global leak. We found that masking attention of the last four layers (21-24) gives the best scores (see Figure 14).

Another important hyperparameter is the variance (σ^2) in the saliency estimation, which is used to determine the degree of global context subtracted from a region (see Equation 4). The higher the variance, the more global context is reduced. As seen in Figure 15, the optimal spot is at $\sigma^2 = 2.5$.

The last parameter is the merging threshold τ_{merge} used to decide which segment candidates to merge (Section 3.4). We found that $\tau_{merge} = 0.8$ returns the best scores on both IoU$_{\text{tt}}$ and IoU$_{\text{st}}$ on the tuning set (see Figure 16).

F. User Study Implementation Details

We conducted a user study using Amazon Mechanical Turk. Each evaluation task contains a detailed instruction and 30 questions. We did not limit the number of tasks per human evaluator. For each question, the evaluators were shown a predicted segment, in the form of a highlighted region, overlaid on an input image, and its predicted text label. The evaluators were then asked to rate how well the label describes the segment on a scale of 0-3, defined in the provided instruction as shown in Figure 21. There were a total of 23,076 questions, each evaluated by three different evaluators. The total number of unique evaluators was 429, and the average number of questions answered by the evaluators was about 155. We calculated the scores (Section 4.3) for each of the three batches separately then reported the average. We include the full instruction and a task example in Figure 21.

G. Additional Results

We present more qualitative results in this section. In Figure 18, we include more results from the ablation experiment in Section 5.3. We show random results of our method in Figure 19 for the Pascal Context dataset and Figure 20 for the Pascal VOC 2012 dataset.

H. Potential Negative Societal Impacts

Unlike traditional segmentation methods, our method outputs arbitrary text labels and may describe people with incorrect assumptions or discriminatory characteristics based on their stereotypical appearances, such as body shape, clothes, nationality, and sexual orientation. For example, we found ‘Asian woman’ or ‘homeless’ in some generated, which can be offensive in some scenarios. Some characteristics, such as beauty and politics, are rather subjective and challenging to filter without human intervention. Due to the data-driven nature of the pre-trained models we use, our model would also be biased toward the culture, preferences, and characteristics of the training sets and may pose controversial issues.

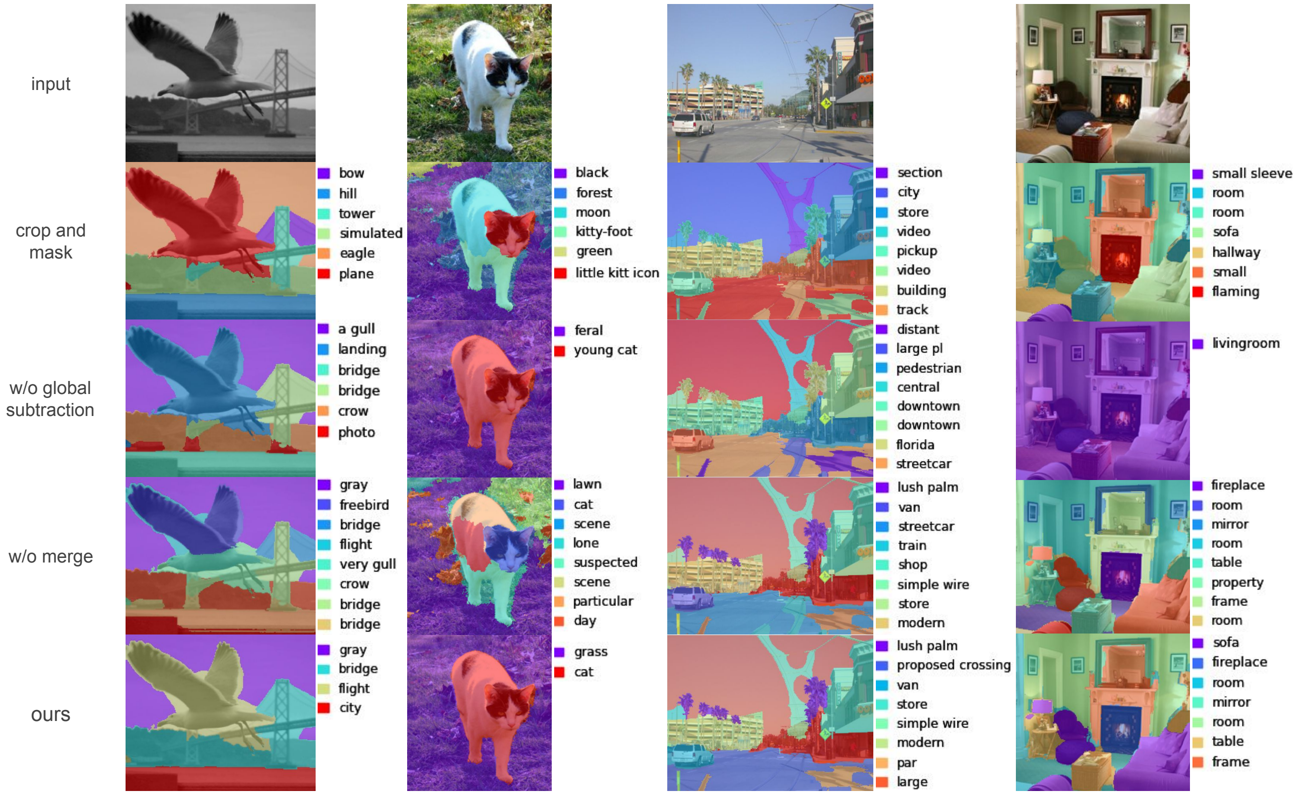


Figure 18. Qualitative ablation analysis

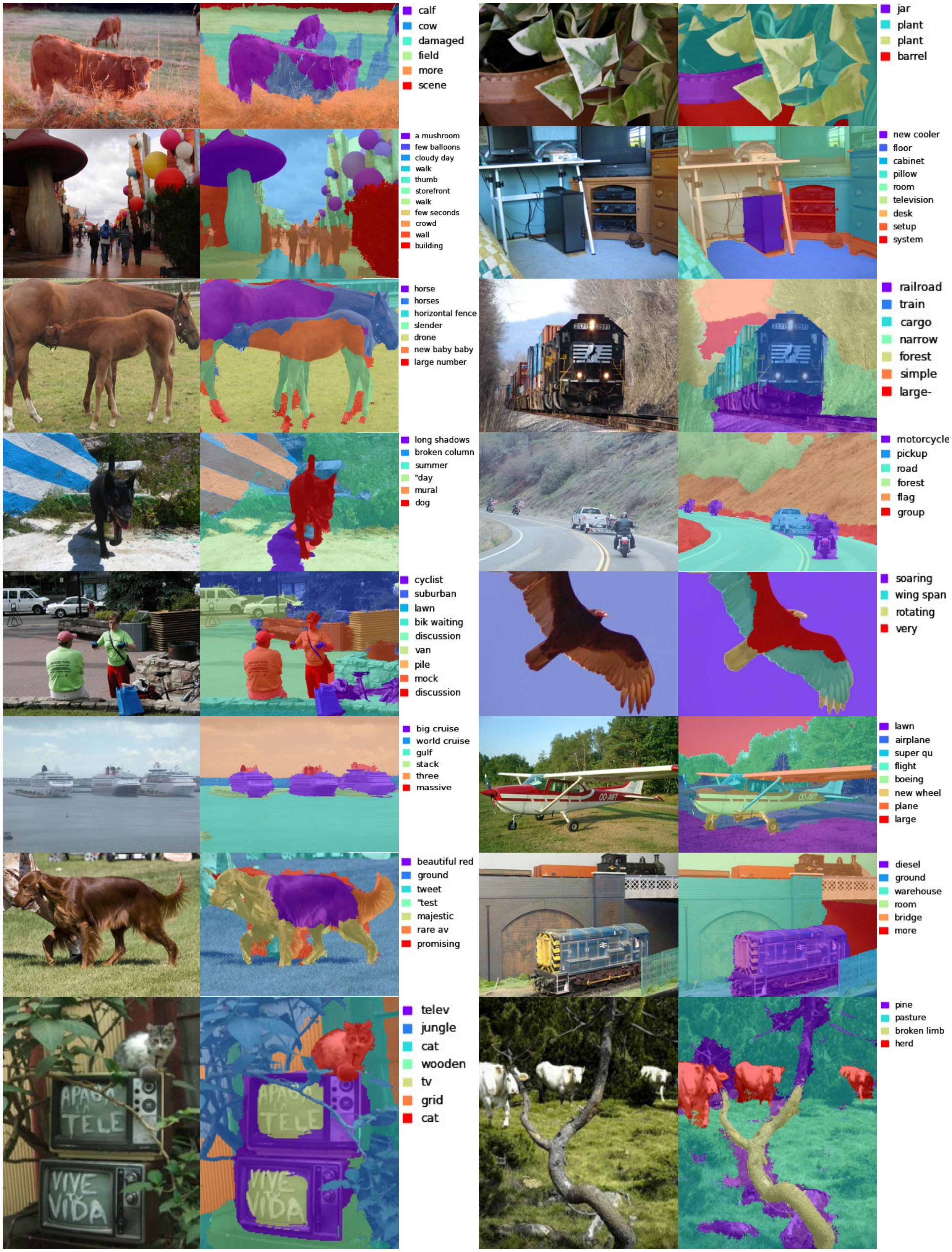


Figure 19. Randomly sampled results from Pascal Context

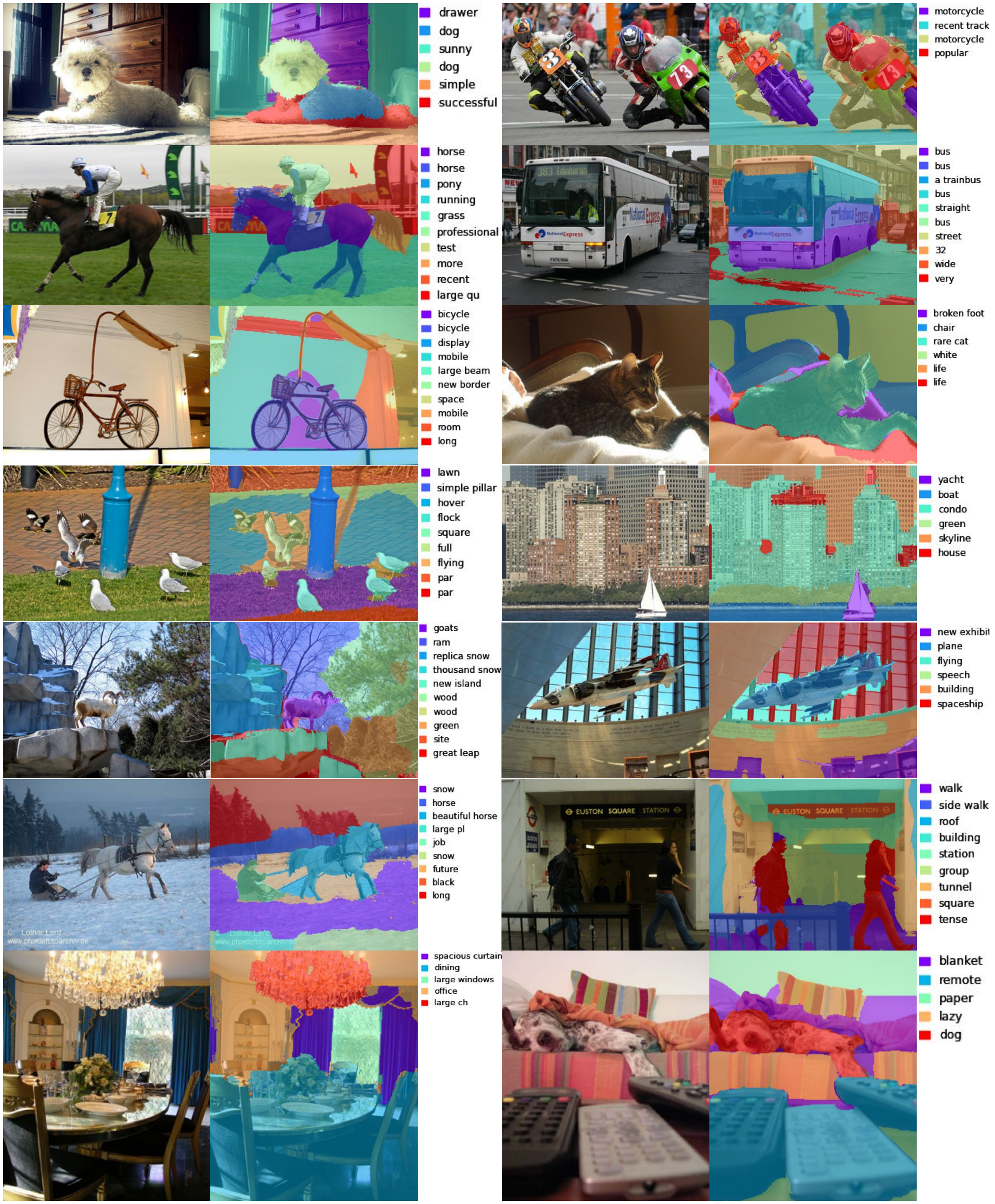
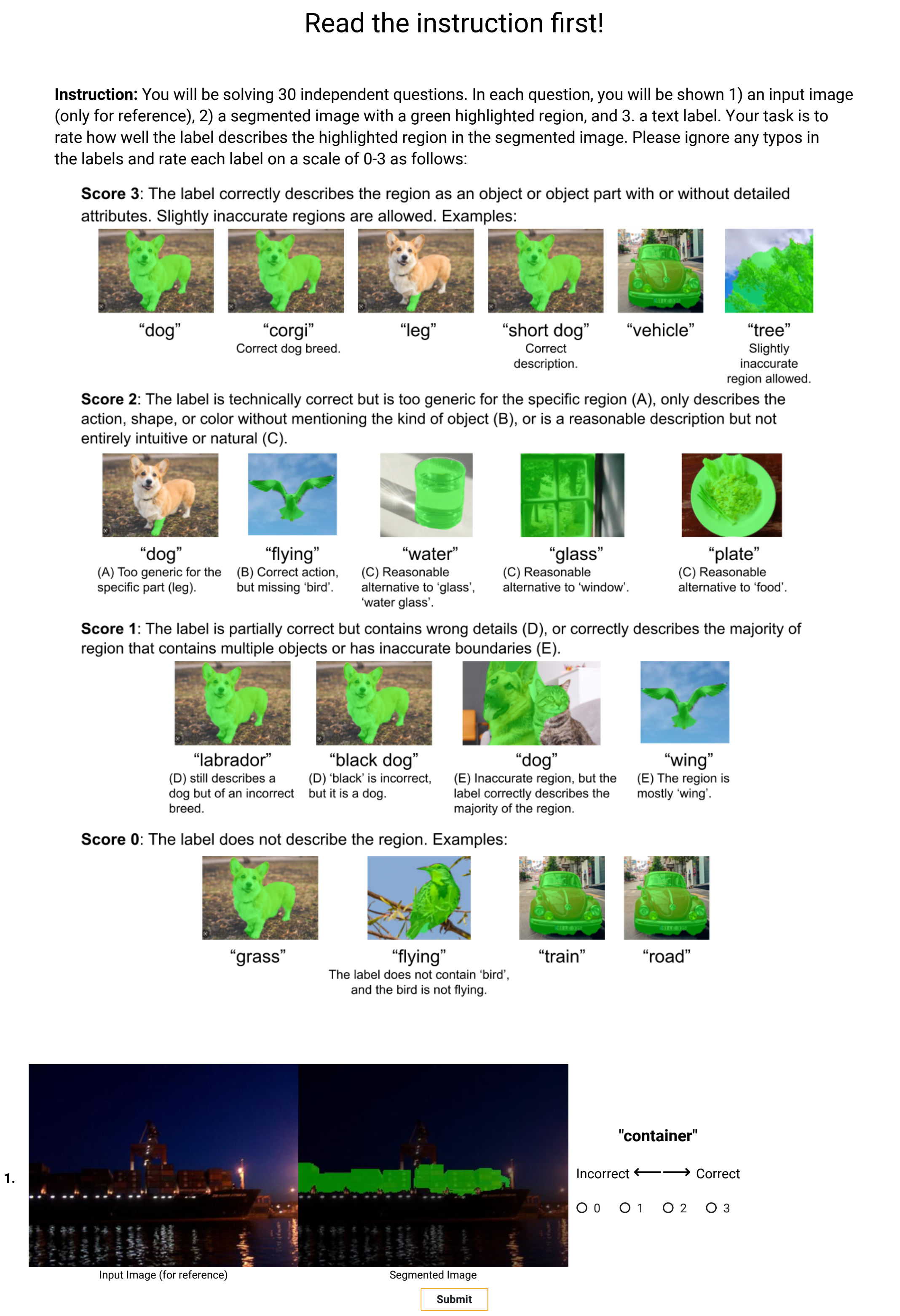


Figure 20. Randomly sampled results from Pascal VOC 2012



Read the instruction first!

Instruction: You will be solving 30 independent questions. In each question, you will be shown 1) an input image (only for reference), 2) a segmented image with a green highlighted region, and 3. a text label. Your task is to rate how well the label describes the highlighted region in the segmented image. Please ignore any typos in the labels and rate each label on a scale of 0-3 as follows:

Score 3: The label correctly describes the region as an object or object part with or without detailed attributes. Slightly inaccurate regions are allowed. Examples:

"dog"

"corgi" Correct dog breed.

"leg"

"short dog" Correct description.

"vehicle"

"tree" Slightly inaccurate region allowed.

Score 2: The label is technically correct but is too generic for the specific region (A), only describes the action, shape, or color without mentioning the kind of object (B), or is a reasonable description but not entirely intuitive or natural (C).

"dog" (A) Too generic for the specific part (leg).

"flying" (B) Correct action, but missing 'bird'.

"water" (C) Reasonable alternative to 'glass', 'water glass'.

"glass" (C) Reasonable alternative to 'window'.

"plate" (C) Reasonable alternative to 'food'.

Score 1: The label is partially correct but contains wrong details (D), or correctly describes the majority of region that contains multiple objects or has inaccurate boundaries (E).

"labrador" (D) still describes a dog but of an incorrect breed.

"black dog" (D) 'black' is incorrect, but it is a dog.

"dog" (E) Inaccurate region, but the label correctly describes the majority of the region.

"wing" (E) The region is mostly 'wing'.

Score 0: The label does not describe the region. Examples:

"grass"

"flying" The label does not contain 'bird', and the bird is not flying.

"train"

"road"

1.

Input Image (for reference)

Segmented Image

"container"

Incorrect ⟵⟶ Correct

○ 0 ○ 1 ○ 2 ○ 3

Submit

Figure 21. User interface for the user study with a full instruction, definitions, and examples of each score.