A. Supplementary Material

First, we show a list of hyper-parameters and implementation details for all of our adaptation methods in Section A.1. Then we show some samples of images and their corresponding per-pixel masks, along with the verification algorithm for counting and occlusions in Section A.2. Then we show some graph samples from our pool of manually designed scenes for the W-VQA dataset and describe their functionality for our automatic triplet (IQA) generation in Section A.3. Finally in Sections A.4 and A.5 we show some samples from W-VQA and H-VQA we randomly select from a diverse set of scenes, with different backgrounds, camera position and illumination.

A.1. Hyper-parameter Selection

The following are all the hyper-parameter selection for all of our algorithms: lr refers to learning rate, E to the number of training epochs, O to the optimizer type, O_{wd} is the optimizer weight decay, O_ϵ is the term added to the denominator to improve numerical stability, O_β are a tuple of coefficients used for computing running averages of gradient and its square. For the Adversarial and MMD methods, the auto-encoder network (AE) is trained separately, in a 2 step format following Zhang et al. [67] Two-stage DA; in both cases the first number in E refers to the training epoch parameter for the AE. For Domain Independent, di_{tokens} is the additional output we use for the synthetic answer tokens.

Adversarial		MMD	
$lr = 15e-4$	$E = 100 + 13$	$lr = 1e-3$	$E = 150 + 13$
$O_{wd} = 1e-6$	$O =$ Adam	$O_{wd} = 1e-4$	$O =$ Adam
$O_\epsilon = 1e-4$	$O_\beta = (0.8, 0.8)$	$O_\epsilon = 1e-4$	$O_\beta = (0.8, 0.8)$
$\alpha = \frac{2}{(1+exp(-10*p))-1}$		$\alpha = 0.4$	$\beta = 0.6$
Domain Independent		**F-SWAP**	
$lr = 15e-4$	$E = 13$	$lr = 15e-4$	$E = 13$
$O_{wd} = 0.2$	$O =$ Adam	$O_{wd} = 1e-1$	$O =$ Adam
$O_\epsilon = 1e-9$	$O_\beta = (0.9, 0.9)$	$O_\epsilon = 1e-9$	$O_\beta = (0.9, 0.98)$
$di_{\text{tokens}} = 100$		$\beta = 1.$	$\lambda = 0.2$

Table 5. Hyper-parameter selection details for all methods.

A.2. RGB and Mask Samples

ThreeDWorld (TDW) [14] allows to capture the RGB images from the camera view along with the id and category per-pixel semantic masks, which we later use to verify the number of objects in the image and avoid object occlusions. Figure 6 shows some samples we randomly select from our generated W-VQA set. The first column correspond to the RGB image, the second and third columns correspond to the category and id masks respectively. We verify if an object overlaps to another and assess the object counts by computing the intersection over union.

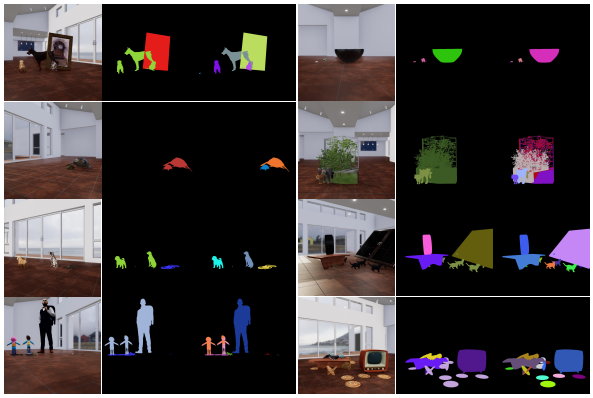

Figure 6. Random samples from the images we generate using TDW along with their category masks (second row) and id masks (third row).

A.3. Scene-Graph Samples

Let E denote the set of scene entities and consider the set of binary relations R. Then a scene graph $SG \in E \times R \times E$ is a collection of ordered triplets (o, p, o) = object, position, and object. For example, as shown in the first sample in Figure 7, with A=lamp, B=table, C=backpack, the triplet $(A, position, B)$ indicates that a lamp is on top of the table, or the table is under the lamp. Similarly, the triplet $(B, position, C)$ indicates that the backpack is to the left of the table, or the table is to the right of the backpack. In this way, from a relationship, there are at least two possible positions, $p \wedge p^{-1}$, e.g., p = left and p^{-1} = right. When sampling from these graphs, each node in E could also be assigned three different attributes: the number of objects to appear in the same scene $n = randrange(20)$, the color, and material type which are selected from a list of available materials and colors from the set of Records in TDW [3].

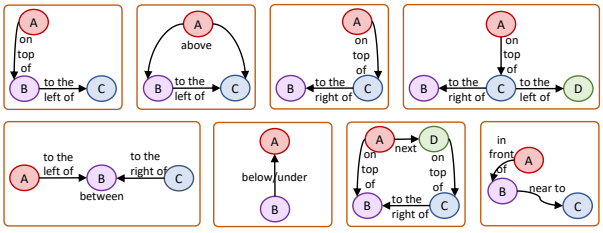

Figure 7. Some of the scene graphs designed for our automated synthetic dataset generation. While generating images, we select one graph and randomly select the number of objects per position $node := [A, B, C, D]$, it's color and materials. Then we use the grammar introduced in Section 3.2 to generate the questions and corresponding answers.

[3] https://github.com/threedworld-mit/tdw/blob/master/Documentation/misc_frontend/materials_textures_colors.md

A.4. W-VQA Generated Samples

We show some random samples we generate for our W-VQA dataset in Figure 8, following Section 3.2.

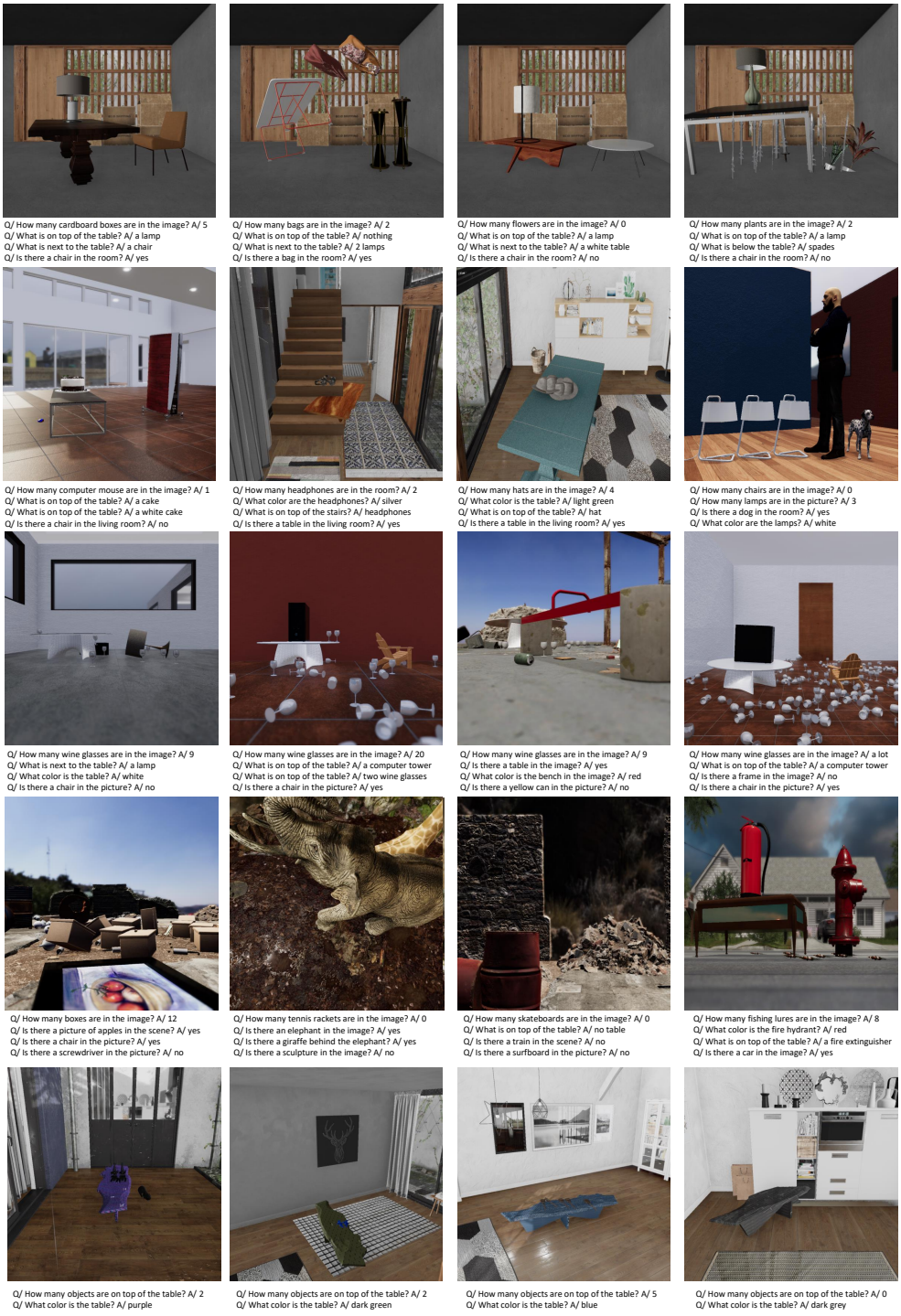

Figure 8. Additional samples of our W-VQA dataset. The first row showcase simple configurations using the same background. The second row shows diverse compositions using indoor scenes. The third row shows compositions of challenging counting questions. The fourth row shows outdoor objects and scenes. Finally, the fifth row shows materials and color related questions using the same object in different camera positions. Best viewed in color.

A.5. H-VQA Generated Samples

We show some random samples we generate for our H-VQA dataset in Figure 9.

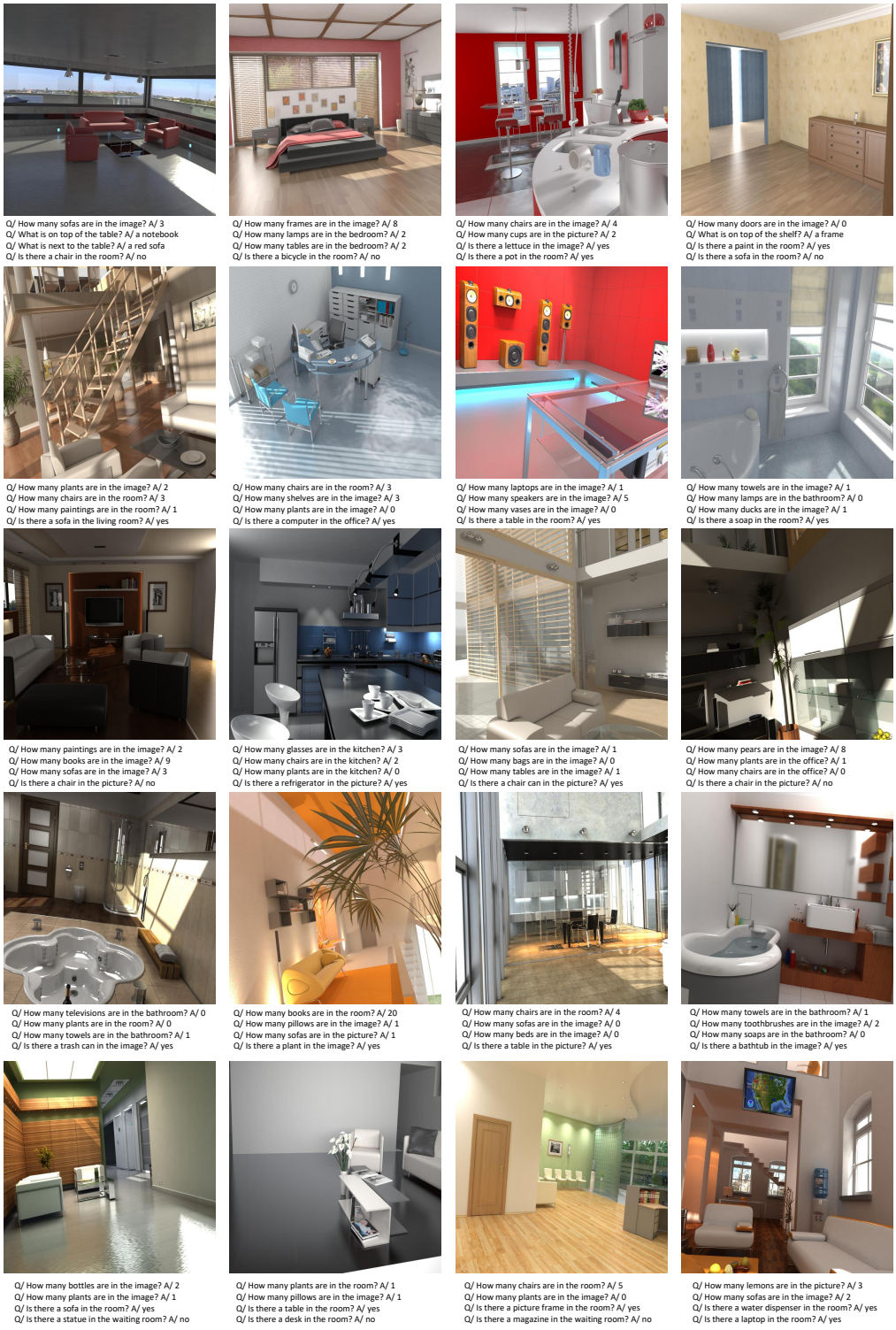

Figure 9. Additional samples of our H-VQA dataset. We generate questions and answers from manual and existing semantic annotations from Hypersim [45]. Best viewed in color.