SimVQA: Exploring Simulated Environments for Visual Question Answering

Paola Cascante-Bonilla[†*] Hui Wu[‡] Letao Wang[♮] Rogerio Feris[‡] Vicente Ordonez[†]
[†]Rice University [‡]MIT-IBM Watson AI Lab [♮]University of Virginia

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-6946-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CVPR52688.2022.00500

Abstract

Existing work on VQA explores data augmentation achieve better generalization by perturbing images in dataset or modifying existing questions and answers. Wh these methods exhibit good performance, the diversity of questions and answers are constrained by the available ages. In this work we explore using synthetic comput generated data to fully control the visual and langua space, allowing us to provide more diverse scenarios. quantify the effectiveness of leveraging synthetic data real-world VQA. By exploiting 3D and physics simulati platforms, we provide a pipeline to generate synthetic da to expand and replace type-specific questions and answ without risking exposure of sensitive or personal data th might be present in real images. We offer a comprehens analysis while expanding existing hyper-realistic datas to be used for VQA. We also propose Feature Swapping (SWAP) – where we randomly switch object-level featu during training to make a VQA model more domain inva ant. We show that F-SWAP is effective for improving VQ models on real images without compromising on their a curacy to answer existing questions in the dataset.

Figure 1. Training samples for VQA from real and synthetic datasets. The first row shows existing examples from the VQA 2.0 dataset. The second row shows examples from Hypersim [45], a hyper-realistic synthetic dataset we extend for VQA. The third row shows some examples we generate using ThreeDWorld [14]. We show type-specific questions for each dataset, i.e., counting questions, color related questions, and yes/no questions.

1. Introduction

Data augmentation is an effective way to achieve better generalization on several visual recognition and natural language understanding tasks. Existing work on Visual Question Answering (VQA) has explored augmenting the pool of questions and answers, e.g. by perturbing or masking some parts of the images [1, 6, 29, 56]. Moreover, curating large-scale datasets is a laborious task and sourcing images is an expensive process that needs to account for practical issues such as copyright and privacy. Augmenting existing datasets with synthetically generated data offers a path to enhance our existing data-driven models at a lower cost.

Our work focuses on leveraging synthetically generated data through the use of modern 3D generated computer graphics using a couple of novel resources – Hypersim [45] and ThreeDWorld [14]. In the past, leveraging synthetic data has proven challenging due the particularly wide domain gap between synthetic images and real images. However, there have been some successes in tasks such as eye gaze estimation [53], embodied agent navigation [10, 49, 50], and autonomous driving [39, 44]. There have also been some synthetic datasets for visual question answering such as CLEVR [26] CLEVRER [64], and VQA Abstract [4]. However these VQA datasets build a closed world that is not designed to generalize to real world images. Remarkably, some recent work has managed to show domain transfer from cartoon images to real images [67], but there is still a limitation on how much could be learned from these existing resources. Our proposed Hypersim-VQA and ThreeDWorld-VQA datasets provide a promising alternative that more realistically captures real world settings and offers a path forward in this direction. Figure 1 shows synthetic image samples along with the VQA 2.0 dataset [18].

*Work partially done while interning at the MIT-IBM Watson AI Lab

978-1-6654-6946-3/22/$31.00 ©2022 IEEE
DOI 10.1109/CVPR52688.2022.00500

Our work also proposes feature swapping (F-SWAP) as a simple yet effective method to augment a currently existing VQA dataset with computer graphics generated examples. Existing methods for domain adaptation rely on the assumption that adaptation can be addressed by making the out-of-domain samples match the distribution of the in-domain samples. However current work often operationalizes this assumption by making the input images themselves look more like the real images e.g. [22, 46, 58]. While there has been success in applying these techniques in domain adaptation for a number scenarios, we claim that perhaps adapting the input space is a harder problem that needs to be solved in order to have effective domain adaptation. Feature Swapping relies instead on swapping random object-level intermediate feature representations. We posit that unless realistic style-transfer is desired from the input domain to the target domain, as long as the two domains are matched at the feature level – domain adaptation can take place. We explain and compare our F-SWAP approach with other methods such as adversarial domain adaptation and demonstrate superior results.

Our contributions can be summarized as follows:

- Dataset generation: We are providing an extension of the Hypersim dataset for VQA, and automatically creating a synthetic VQA dataset using ThreeDWorld.
- Feature swapping (F-SWAP): We propose a surprisingly simple yet effective new technique for incorporating synthetic images in our training while mitigating domain shift. Our method does not rely on GANs or adversarial losses which could be difficult to train.
- Experimental results: We provide an empirical analysis, using well known techniques such as adversarial augmentation, domain independent fusion, and maximum mean discrepancy matching to alleviate the visual domain gap vs our proposed approach – and analysis on knowledge transfer between skills.

We first introduce related work (Sec. 2), then we describe our proposed synthetic dataset generation process (Sec. 3), then we explain the motivation and details of our feature swapping method (Sec. 4), then we describe and discuss our experiments(Sec. 5), and finally we conclude the paper (Sec. 6). Our synthetic datasets and code are available at https://simvqa.github.io.

2. Related Work

Our work is related to both general efforts at improving visually-grounded question-answering models, and efforts targeting data augmentation for VQA and the use of synthetic data for other visual reasoning tasks.

Visual Question Answering (VQA). There has been much progress on the task of VQA, where the goal is to answer a question conditioned on both an image and a question text input [4, 42]. A lot of work in VQA measure progress using the VQA 2.0 benchmark [18]. Much of the work in this area focuses on exploring new architectural designs which can effectively model the interaction between the image and the text modalities, such as bilinear pooling [13], bottom-up-top-down attention [2], neural module networks [23, 24], and most recently transformer architectures [9, 34, 35]. However, most work assumes that models are trained on real image-question-answer triplets, and that they will be applied to settings with similar data distributions. Our paper instead investigates a setting where we leverage synthetic training data to learn certain skills so that the model generalizes to real images at test time.

Data augmentation for VQA. Data augmention in VQA has often been studied within the context of model robustness. Chen et al [8] select visual objects in images and words in questions which are critical for answer prediction and synthesizes new samples by masking out critical visual regions or words. Whitehead et al [62] leverage existing linguistics resources to create word substitution rules for paraphrases, synonyms and antonyms which are then used to generate question perturbations for VQA. Gokhale et al [17] explored VQA data synthesis via a combination of semantic manipulation on image content and questions. However, previous work in this direction generates new samples via perturbations on top of the original real-image VQA dataset, which limits the diversity and the range of variations for generated samples. In contrast, our work leverages photo-realistic, multi-physics synthetic environments, and is able to generate rich image-question pairs parameterized by scene/room types, camera view, object dimensions, object counts and object orientations. In concurrent work Gupta et al [20] propose feature swapping for avoiding contextual bias in visual question answering.

Synthetic data using simulated environments. The value of leveraging simulated environments to augment training has been explored in various vision tasks, such as object detection, semantic segmentation, and pose estimation [28, 32, 37, 48, 57, 66]. Synthetic environments have also been applied to vision and language problems, such as embodied agent learning [11, 12, 16, 30, 51, 55], using platforms such as the Unreal Engine [36, 40], and using existing scenes and spaces manually created by specialized designers and content creators [60]. Within the task of VQA, to train and diagnose model performance on compositional questions, synthetic datasets such as CLEVR [26] and CLEVRER [64] have been proposed. However, models trained on such synthetic datasets typically do not generalize to real images as they were designed under a closed world assumption. In this paper, we explore approaches which leverage both real-image VQA data for its richness in visual concepts and question types, as well as synthetic

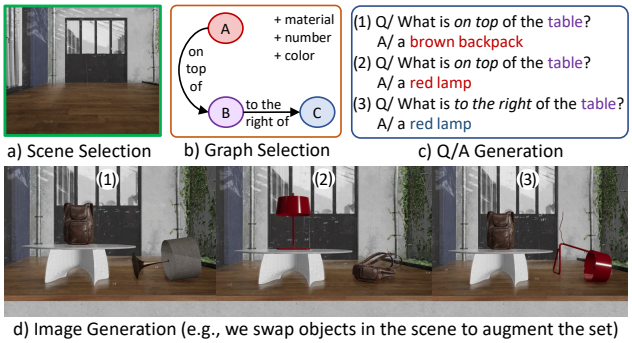

Figure 2. Sample pipeline for generating VQA data using ThreeDWorld. a) Manually select scenes from a set of random camera walks. b) Select one of the generated scene graphs containing object information such as positions, number, color, and materials. c) Generate question-answer pairs following a template based on the scene graph. d) Finally, generate images by placing objects and modifying characteristics of the scene based on steps b and c.

datasets generated from controllable, configurable 3D environment. We found that using this approach we can generate arbitrarily large-amounts of high-quality data for type-specific questions.

3. Synthetic Dataset Generation

First, we describe the generation of a VQA dataset by extending the existing Hypersim dataset [45] (section 3.1). We name this dataset Hypersim-VQA, or H-VQA, for short. Then we explore the automatic creation of a VQA Dataset using ThreeDWorld [14] (section 3.2). We name this dataset ThreeDWorld-VQA, or W-VQA, for short.

3.1. Extending Hypersim for VQA

Hypersim [45] is an existing 3D graphics generated dataset with a high image quality and displays a diverse array of scenes and objects. Hypersim metadata includes the complete geometry information per scene, dense per-pixel semantic instance segmentations for every image, and instance-level NYU40 labels annotations. We extend these data by manually annotating objects on all images given their dimensions and positions in the scene. Additionally, we add questions and answers based on the number of appearances of an object in an image and their location with respect to other objects in the same frame. Since we have the 3D bounding boxes coordinates for each object in a scene, we can calculate the distance d, plunge p and azimuth a for two objects located in positions (x_1, y_1, z_1) and (x_2, y_2, z_2) as

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \quad (1)$$

$$p = \deg\left(\arcsin\left(\frac{z_2 - z_1}{d}\right)\right), \quad (2)$$

$$a = \deg\left(\arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right), \quad (3)$$

Figure 3. In our VQA dataset generation pipeline, we can automatically manipulate the scene composition, object materials and colors, allowing our grammar to generate more challenging questions and answers.

where d is the amount of space between two objects, p is the angle of inclination measured from the horizontal axis formed by aligning two objects, and a is the angle between two objects, measured clockwise with respect to the North. Once we have the distance between all objects in a scene, we define clusters of objects and use the azimuth and plunge to define the position of one object with respect to the other objects in the same scene. Finally, we generate yes/no questions and answer pairs based in the visibility of an object in a scene frame. We call this new set Hypersim-VQA.

3.2. Automatic VQA Generation

ThreeDWorld (TDW) [14], is a platform for interactive multi-modal physical simulation that we use to generate images. We follow the steps shown in Figure 2 to generate the image I, question Q and answer A triplets for our W-VQA dataset. In this section, we provide a detailed description of the synthetic-generation pipeline.

TDW Model Library TDW contains $2,323$ objects, 585 different materials, and 44 scenes with 35 scenes indoors and 9 outdoors. This variety of assets give us a significant level of freedom to generate diverse and challenging image compositions and question/answer pairs. Using the TDW ModelLibrarian[1], we have access to a model asset bounding points which we use to calculate the volume of an object. Using volume information we assign dimension-related categories (e.g., tiny, small, mid-range, large, etc) to each object. We also have access to a set of category labels, that correspond to ImageNet labels, already assigned to each object asset (3D objects). Additionally, we manually annotate the color of each material asset, and added detailed descriptions for each object in the available TDW model set.

Scene graph generation We manually designed a set of simple scene graphs [27, 54] that describe objects, relation-

[1]https://github.com/threedworld-mit/tdw

ships between objects, and the attributes of each object. We assign the position relationship of these objects given the available range of sizes and object model categories, e.g., small and tiny could be on top of tables, chairs, or other large objects, and large objects could be placed in the vicinity of another large object. We showcase an example of this in Figure 2. The number of objects in a scene is randomly selected based on the object size, e.g., we could place dozens of small objects that fit in a camera view, but having dozens of large objects may create occlusions and collisions in the image, and some objects may fall outside the camera view. Finally, the color and material attributes are assigned randomly, but we also generate a large set in which we keep most of the object models with their original attribute values, as we further describe in Section 5.1.

Synthetic Image Generation We placed multiple cameras with random configurations in the 44 scenes, by randomly generating x_c, y_c, and z_c coordinates for camera positions, and θ_c for directions that the cameras look at. We then manually selected a set of camera configurations that have good views of an empty room, to later place objects in front of them. For example, Figure 2 illustrates a scene in which we placed a random table at the left of the image, a random small object (backpack or lamp) on the table, and another random small object on the ground, which follows the scene graph depicted in Step B of the Figure. We also change the material of objects at this stage, and place a random number of objects in the image following the scene graph configuration. An example of how changing materials and colors visually affects a generated image is showcased in Figure 3. We calculated the positions of these objects relative to the camera using

$$x = x_c + r\cos\theta, \quad (4)$$

$$y = y_0 + h, \quad (5)$$

$$z = z_c + r\sin\theta, \quad (6)$$

where x, y and z are the position coordinates of the placed object, θ is the direction of the object with respect to the camera, typically within 30 degrees to θ_c, the direction that the camera looks at, r is the distance between the object and the camera, and y_0 is the coordinate of height at the floor level in the scene. In addition, since h is an estimated height with respect to the object size, we waited 25 frames for these objects to fall to their natural stationary positions using the TDW physics engine. Finally, TDW 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.

Question/Answer Generation Questions and answers are generated following a template based grammar associated with a predefined scene graph and it's corresponding image. We show in Figure 4 the template based grammar we use to generate the question/answer pairs. In our setup, a *noun* is directly associated with the model object label category from the TDW asset, *position* is taken from the relationship between objects from the scene graph, and *number*, *adjective_color* and *adjective_material* are taken from the attributes selected when generating the graph and the synthetic image. We refer to the distribution of generated questions and answers with more detail in Section 5.1.

```
count_question ::= ctype_1 | ctype_2
count_answer ::= number

yes_no_question ::= etype_1 | etype_2 | etype_3 | etype_4
yes_no_answer ::= “yes” | “no”

color_question ::= "What is the color of the" noun_q "?"
color_answer ::= adjective_color

noun_question ::= “What is” position “of the” noun_q “?”
noun_answer ::= noun_q

position_question ::= "Where is the" noun_q "?"
position_answer ::= position_q "of the" noun_q "?"

ctype_1 ::= "How many" noun_q "are in the" place_q "?"
ctype_2 ::= "How many" noun_q "are" position "of the" noun "?"

etype_1 ::= "Is there a" noun_q "in the" place_q "?"
etype_2 ::= "Is the" noun "made of" adjective_material "?"
etype_3 ::= "Is the" noun adjective_color "?"
etype_4 ::= "Is the" noun_q position "of the" noun_q "?"

noun_q ::= noun | adjective_material noun | adjective_color noun
place_q ::= place | "image" | "picture"
place ::= "bedroom" | "living room" | "studio" | "building" | "street" | ...
noun ::= "chair" | "person" | "elephant" | "fire hydrant" | "drum set" | ...
position ::= "to the right" | "to the left" | "on top" | "below" | ...
adjective_material ::= "wood" | "metal" | "plastic" | "cotton" | ...
adjective_color ::= "black" | "brown" | "pink" | "blue" | "green" | ...
number ::= "0" | "1" | "2" | "3" | "4" | "5" | ...
```

Figure 4. Template based grammar we use to generate question/answer pairs given our generated image and it's corresponding scene-graph.

4. Feature Swapping

Given a triplet of images I, questions Q and answers A, we have access to three datasets from different domains, where $(I_R, Q_R, A_R) \in R$ correspond to a Real-VQA dataset consisting of real images (we use VQA 2.0 [18]), $(I_H, Q_H, A_H) \in H$ correspond to the Hypersim-VQA dataset, and $(I_W, Q_W, A_W) \in W$ correspond to the TDW-VQA dataset. We assume that the images and their corresponding questions are inputs to a VQA model, and the objective is to predict as output the corresponding ground-truth answers. Our goal with feature swapping is to train a model that is generic and as domain invariant as possible. The motivation for feature swapping relies in observing that in all three datasets we can find similar types of objects and configurations but the appearance of the objects might differ. Our goal with feature swapping is then to randomly replace during the training the object-level features for some of the objects with the features for an equivalent object from another domain.

Given an image I, we use a pre-trained model

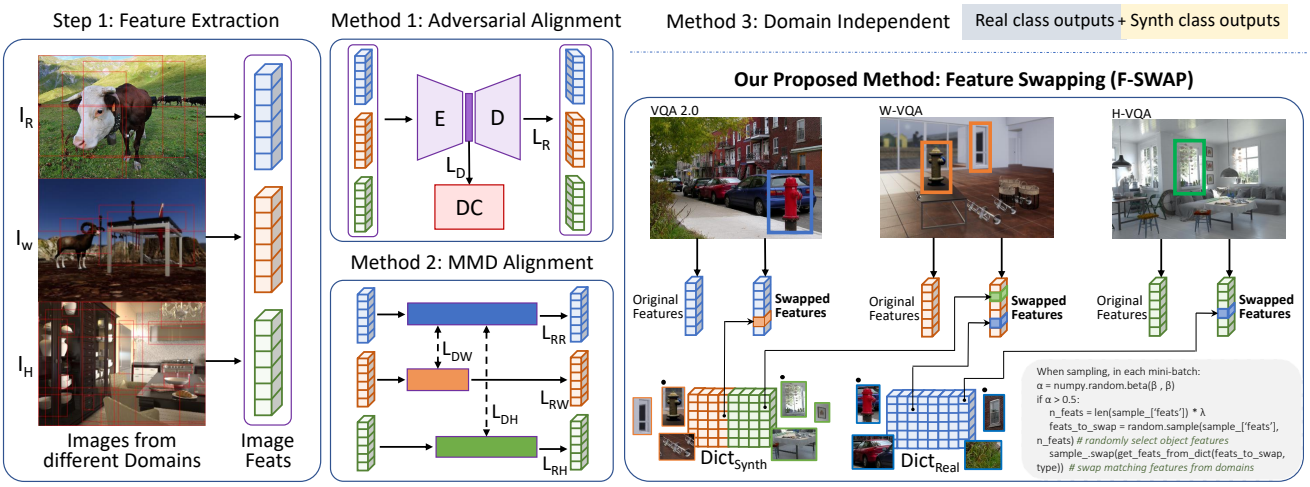

Figure 5. Overview of our training pipeline. First, we extract image the features using Faster-RCNN or CLIP. Then, we use different methods to alleviate the domain gap between real and synthetic images. Methods 1 & 2 yield a set of aligned features and Method 3 augment the output space of the VQA model (i.e., answer tokens), separating real class output tokens and synthetic class output tokens. Our proposed method (F-SWAP) swaps object-level features between domains, which are then used to train the VQA model.

G to extract the image region features $G_f(I) = \{f_1, f_2, ..., f_n\}$ along with their corresponding pseudo-labels $G_{sl}(I) = \{sl_1, sl_2, ..., sl_n\}$ which are labels predicted by Faster-RCNN corresponding to annotations from Visual Genome [31]. Pseudo-labeling has proven effective in semi-supervised learning where only a portion of the training data is annotated [5, 7, 33]. Since we have access to all images from the three sets, we create a dictionary D_{type} per dataset with $type = R \vee H \vee W$, where the $[key, value]$ of a dictionary D_{type} corresponds to the pseudo-label $(sl_i)_{type}$, and all the region features $[(f_i)_{type}, ..., (f_m)_{type}]$ that the model G assign as $(sl_i)_{type}$ respectively. Once we retrieve the information for all the dictionaries D_R, D_H, D_W, we use them to swap features from one dataset to the other. While training, when sampling datapoints from R, we randomly select an Image I_R and get all the region features $G_f(I_R)$ and its corresponding pseudo-labels $G_{sl}(I_R)$. Since we have access to all dictionaries, we lookup for the pseudo-labels that also exist in $D_H \vee D_W$, for simplicity, $D_S = D_H \vee D_W$, thus, after obtaining $G_{sl}(I_R) \in D_S$ we proceed to randomly select a portion $\lambda |G_f(I_R)|$ of the corresponding pseudo-labeled features in D_S and replace them with the matching features in I_R. In all of our experiments, $\lambda = 0.2$. Figure 5 shows pseudo-code for this algorithm.

In VQA, the model takes as input pre-computed region features that come from a pretrained object detection model, but the prediction scores are often ignored. VQA models assume that these region features contain enough information for vision and language reasoning. Here we are assuming that the pseudo-labels associated with region features are good predictions; thus, we are augmenting the feature space of the input images from the real domain with features from the synthetic domain by perturbing the input image feature space with features that G scores as similar. In contrast to other methods that rely on adversarial augmentation, our presented approach does not need training or any adaptation. Importantly, since we are performing feature replacements in latent representations we also bypass the need to perform anything resembling style-transfer in the input pixel domain.

5. Experiments

First, we describe our experimental settings (Sec. 5.1), then we describe our data augmentation experiments (Sec. 5.2), then we describe our experiments using various domain alignment techniques including F-SWAP (Sec. 5.3), and finally we show how certain types of question beyond counting can influence the accuracy of counting questions (Sec. 5.4).

5.1. Experimental Settings

Datasets. **Real-VQA.** Following Whitehead et al [63]'s skill-concept separation for compositional analysis, we take the VQA 2.0 dataset [18], and separate the counting questions for a detailed analysis on how synthetic data may affect a model performance. For training, we create two different splits: R-VQA$_C$ that corresponds to the training set with only counting questions, and R-VQA$_{NC}$ which corresponds to the VQA 2.0 training set without counting questions. R-VQA$_C$ contains $48,431$ datapoints, and R-VQA$_{NC}$ contains $378,018$ datapoints. For testing, we use the standard VQA 2.0 validation set and report our results on *Numeric* questions, where $\sim 85\%$ of the questions correspond to counting questions, *Others*, and *Overall* for the general accuracy. **Hypersim-VQA.** The Hy-

persim dataset [45] comes with annotations corresponding to the NYU40 labels. Additionally, we manually annotate 460 scenes and add 1,250 extra labels for objects whose semantic masks has generic annotations, such as *otherstructure*, *otherfurniture* and *otherprop*. We then generate 254,174 counting questions for 41,551 images. In our experiments, we use a subset of 20,000 questions that only containe NYU40 labels (excluding *otherstructure*, *otherfurniture* and *otherprop*) and include 10,000 randomly selected from the extra annotated labels. We also generate 40,000 yes/no questions probing whether an object is present in an image following Section 3.1. **TDW-VQA.** We generate 33,264 counting related datapoints and add 30,000 yes/no questions to the same images. Additionally, we generate 12,000 extra images and add color and material questions following Section 3.2, for a total of 87,264 automatically generated datapoints using the ThreeDWorld simulation platform.

Base VQA model. Multi-modal transformer-based architectures currently hold the state-of-the-art results on VQA Challenge [9, 34, 65]. For our experiments, we select the top-performing model without large-scale pre-training [65] as our base model. Our base code follows the hyperparameter selection included in their publicly available implementation[2].

5.1.1 Image Features.

We experiment with three types of input image features:

Region Features. We extract intermediate features from a Faster R-CNN model [43] with ResNet-101 as the backbone, pretrained on the Visual Genome dataset. Following [3, 65], we obtain a dynamic number of objects $m \in [10, 100]$ by setting a confidence threshold. If the number of objects is lower than 100, we use zero-padding to fill the final matrix of shape 100×2048.

Grid Features from CLIP ResNet-50. Following [25, 52], we use the CLIP model [41] with the ResNet-50 visual backbone and extract the features from the RoI Pooling layer without any additional fine-tuning. With this approach, we can extract an image representation matrix of size 558×2048.

Grid Features from CLIP ViT-B. We also divide the raw input images into grids of 2×2, 4×4 and 8×8, and use them as inputs for the CLIP ViT-B model [41]. We then aggregate the outputs and we use them as an image representation matrix of size 85×552.

[2]https://github.com/MILVLG/mcan-vqa

Feature backbone	Feature size	Training data: Real	Training data: Synthetic		R-VQA Accuracy
		R	H	W	*Numeric*
FRCNN – RN101	100×2048	✓			42.73
		✓	✓		44.70 +1.97
		✓		✓	42.86 +0.13
CLIP - RN50	558×2048	✓			42.83
		✓	✓		43.61 +0.78
		✓		✓	42.91 +0.08
CLIP – ViT-B	85×512	✓			41.93
		✓	✓		43.98 +2.05
		✓		✓	41.35 −0.58

Table 1. Data augmentation using synthetic data improves Real-VQA performance (R-VQA) on numeric questions, especially when using Hypersim-VQA (H). In all these experiments only counting questions were used for training from both the existing Real-VQA dataset, VQA_C (**R**) and our synthetic dataset variants: Hypersim-VQA (**H**) and TDW-VQA (**W**).

5.1.2 Textual Features.

Following [65] we tokenize the input questions into words and transform them into feature vectors using pre-trained 300-dimensional GloVe word embeddings [38]. These embeddings are passed through a one-layer LSTM [21]. We then use all the output features for all corresponding words.

5.1.3 Domain alignment methods.

For comparisons to earlier work, we consider the following domain alignment methods that have been proposed in the past either for VQA or for other similar visual recognition problems.

Adversarial adaptation. This approach is a modification of the unsupervised domain adaptation of Ganin et al [15]. Since our goal is to minimize the feature gap from real I_R and synthetic $(I_W \cup I_H)$ images, instead of using the questions or answers ground-truth to predict the class labels (as the label predictor block), we use an auto-encoder $D(E(\cdot))$ to reconstruct the input features X of the images, and a domain classification model DC that is trained to distinguish the domain of each input. This domain classifier is then connected to the underlying input features X but its gradients are multiplied by a negative constant during training. This gradient reversal layer encourages the features of both domains to remain indistinguishable. This process commonly known as adversarial domain adaptation is optimized in an alternative fashion as follows:

$$L_D = \sum (\log(DC(X)) + \log(1 - \hat{DC}(X))), \quad (7)$$

$$L_R = \sum (D(E(X)) - \hat{D}(\hat{E}(X)))^2, \quad (8)$$

$$L_{total} = L_R + \alpha L_D, \quad (9)$$

where L_{total} is the loss function to be optimized that encourages a good reconstruction while discouraging the features to encode any domain specific information.

Feature backbone	Feature size	Training data: Real	Training data: Synthetic		R-VQA Accuracy		
		R-VQA$_{NC}$	H-VQA$_C$	W-VQA$_C$	*Numeric*	*Others*	*Overall*
FasterRCNN – RN101	100×2048	✓			6.08	68.94	60.69
		✓	✓		15.99$_{+9.91}$	68.97	62.02$_{+1.33}$
		✓		✓	21.18$_{+15.1}$	68.91	62.65$_{+1.96}$
		✓	✓	✓	24.96$_{+18.8}$	68.91	63.14$_{+2.45}$
CLIP - RN50	558×2048	✓			4.55	69.70	61.15
		✓	✓		10.24$_{+5.69}$	69.63	61.84$_{+0.69}$
		✓		✓	14.67$_{+10.12}$	69.45	61.76$_{+0.61}$
		✓	✓	✓	17.81$_{+13.26}$	69.81	62.98$_{+1.83}$
CLIP – ViT-B	85×512	✓			5.06	70.12	61.58
		✓	✓		14.06$_{+9.00}$	70.06	62.71$_{+1.13}$
		✓		✓	17.25$_{+12.19}$	70.06	63.12$_{+1.54}$
		✓	✓	✓	20.72$_{+15.66}$	70.05	63.57$_{+1.99}$

Table 2. Learning counting skill on real data using data augmentation with our synthetic datasets. In all these experiments only counting questions were used from our synthetic dataset variants: Hypersim-VQA (H-VQA$_C$) and TDW-VQA (W-VQA$_C$).

Distribution alignment adaptation. In this approach, we use N auto-encoder architectures $D(E(\cdot))$ corresponding to the datasets we want to align, (e.g., VQA 2.0 as R, TDW as W, and Hypersim as H) we then compute the Maximum Mean Discrepancy (MMD) loss [19] among the intermediate layers of each model. By doing this, the real and synthetic features distributions are encouraged to get closer as similarly used for domain adaptation in [47, 59]. This is performed as follows:

$$L_{D\diamond} = \text{MMD}(E(X_R), \hat{E}(X_\diamond)), \tag{10}$$

$$L_{R\diamond} = \sum (D(E(X_\diamond)) - \hat{D}(\hat{E}(X_\diamond)))^2, \tag{11}$$

$$L_{total} = L_R + \alpha L_{DW} + \beta L_{DH}, \tag{12}$$

where $\diamond$ can be replaced by W for the TDW features, and H for our extended Hypersim dataset features, and R represents Real-VQA features. Unlike the adversarial domain adaptation approach, here the adversary is not a classifier but a loss that tries to match the distribution of the features across the pair of domains.

Domain independent fusion. Inspired by Wang et al [61]'s work on bias mitigation, we perform domain independent training, where we treat the real and synthetic output space as separate. To do so, we create a new set of classes that contains tokens from the synthetic set only, and extend the real set answer token space with these new tokens, as show in the third method of Figure 5. This approach can be viewed as two classifiers with a shared backbone that has access to the decision boundary of both the real and synthetic domain.

5.2. Data augmentation experiments

First, we evaluate the effect of augmenting Real-VQA data with the proposed synthetic datasets. We are interested to test if the ability of VQA models to answer counting questions on synthetic data could improve the counting performance on real VQA data. We experiment with two different settings for data augmentation. The first setting tests a scenario where real and synthetic data contain the same question type (in this case, counting questions). Table 1 shows that, under different feature backbones, the performance of counting questions on real data is improved when R-VQA$_C$ is augmented with the proposed H-VQA dataset.

The second setting targets a more challenging case, where the real data does not overlap with the synthetic data in terms of questions types. Specifically, in this setting, for real data, we use R-VQA$_{NC}$, which does not contain counting questions. So the model needs to learn the skill for counting questions from the augmented synthetic data alone. Table 2 shows that in all settings of feature backbones, and different combinations of synthetic data augmentations, the model learns to answer counting questions. In this case, augmenting with ThreeDWorld-VQA seems to outperform augmenting with Hypersim-VQA, perhaps due to a greater extent of controllability of the generated scenes. Lastly, the best results are obtained by data augmentation using both synthetic datasets.

5.3. Domain alignment.

As demonstrated in Section 5.2, counting skills learned from our synthetic datasets can effectively transfer to real VQA data, even when the real training data does not contain counting questions. Here, we explore to what extent skill learning using synthetic data can be helped by explicit alignment of visual features between two domains. The real data used in this experiment includes R-VQA$_{NC}$, as well as R-VQA$_C$ under three different regimes (0%, 1%, 10%).

Table 3 summarizes the experimental results when using

Data	Method	+0% R-VQA$_C$			+1% R-VQA$_C$			+10% R-VQA$_C$		
		Numeric	*Others*	*Overall*	*Numeric*	*Others*	*Overall*	*Numeric*	*Others*	*Overall*
H-VQA$_C$	Simple Augmentation	15.99	68.97	62.02	29.64	68.45	63.34	35.72	68.61	64.29
H-VQA$_C$	Adversarial	16.07$_{+0.08}$	66.01$_{-2.96}$	59.46$_{-2.56}$	28.31$_{-1.33}$	66.89$_{-1.56}$	61.83$_{-1.51}$	35.01$_{-0.71}$	66.91$_{-1.7}$	62.71$_{-1.58}$
H-VQA$_C$	MMD	24.79$_{+8.80}$	67.13$_{-1.84}$	61.58$_{-0.44}$	31.61$_{+1.97}$	67.78$_{-0.67}$	63.04$_{-0.30}$	38.87$_{+3.15}$	68.36$_{-0.25}$	64.49$_{+0.2}$
H-VQA$_C$	Domain Independent	22.87$_{+6.88}$	68.65$_{-0.32}$	62.64$_{+0.62}$	29.05$_{-0.59}$	68.73$_{+0.28}$	63.52$_{+0.18}$	37.67$_{+1.95}$	69.34$_{+0.73}$	65.17$_{+0.88}$
H-VQA$_C$	Feature Swapping (F-SWAP)	23.38$_{+7.39}$	69.07$_{+0.10}$	63.07$_{+1.05}$	31.64$_{+2.00}$	69.08$_{+0.63}$	64.15$_{+0.81}$	39.71$_{+3.99}$	69.13$_{+0.52}$	65.26$_{+0.97}$
W-VQA$_C$	Simple Augmentation	21.18	68.91	62.65	31.18	68.97	64.01	38.47	68.86	64.87
W-VQA$_C$	Feature Swapping (F-SWAP)	26.84$_{+5.66}$	68.89$_{-0.02}$	63.67$_{+1.02}$	31.21$_{+0.03}$	68.82$_{-0.15}$	63.89$_{-0.12}$	38.54$_{+0.07}$	68.97$_{+0.11}$	64.97$_{+0.10}$

Table 3. Counting skill learning under different low-regime settings for Real VQA counting questions (R-VQA$_C$). All models share the basic training set: VQA$_{NC}$ (the non-counting subset of VQA v2 training data).

different domain alignment approaches. Compared to the baseline method of simple data augmentation, we do not observe an overall improvement with Adversarial Adaptation. Compared to Domain Independent, MMD seems to generate more consistent gains on counting questions, across various regimes for R-VQA$_C$; however, this gain is also accompanied by decreased performance on the split of *Others* and sometimes on the overall evaluation data. Finally, the results suggest that Feature Swapping outperforms the baseline and other domain alignment methods, and produces consistent gains on counting questions as well as the overall accuracy, across different regimes of VQA$_C$.

5.4. Effect of question distribution.

In previous experiments, we focus on augmenting the real dataset with synthetic data of a specific skill type. In this section, we experiment with increased diversity of questions on synthetic data, and how it may effect the performance on different subsets of the real data. As shown in Table 4, on the *Others* category, we observe increased performance when adding more question types with the TDW-VQA dataset but not with Hypersim-VQA, likely due to the richer object repository and the more controllable environment of TDW. Interestingly, for both datasets, adding other question types results in a noticeable gain on the counting questions. We hypothesize that these additional questions help with visual concept learning (on color, object existence, etc), which consequently benefits counting skill learning since visual concept learning is a basic step to answering counting questions.

6. Conclusion

In this paper we demonstrated the efficacy of VQA datasets generated using 3D computer graphics to incorporate new skills into existing VQA models trained on real data. We particularly showed that we can teach a VQA model how to count objects in the real world by using only synthetic data while not decreasing the model performance on other types of questions. This is challenging since real and synthetic datasets often exhibit a large domain gap. We further proposed F-SWAP as a simple yet effective technique for domain adaptation that is competitive and surpasses previous methods in our experiments.

Training data: R-VQA$_{NC}$	R-VQA Accuracy		
	Numeric	*Others*	*Overall*
H	15.99	68.97	62.02
H + Yes/No Questions	22.11	68.38	63.17
W	21.18	68.91	62.65
W + Yes/No Questions	25.43	70.10	64.24
W + Color Questions	26.98	70.24	64.56

Table 4. Effect of the distribution of synthetic data. We add other type-specific questions to our synthetic data and evaluate its effect on real data.

7. Broader Impact

The main ethical aspects of this work have to do with data privacy and mitigating implicit biases in the existing VQA models. In this work, we explored 3D simulation platforms to generate realistic synthetic data as a promising direction to augment or replace existing datasets, effectively avoiding the exposure of potentially sensitive information. However, our approach is not able to generate a broad diversity of animated objects (e.g., people or animals interacting in the scenes) given that Hypersim only contains indoor scenes, and TDW provides a limited quantity of these type of model assets. While we leverage this information in our generated data, a more complex 3D system would be ideal to craft a more diverse set of animated objects. Therefore, this work is a stepping stone for further explorations to address this issue in the future.

Acknowledgments This work was supported in part by the National Science Foundation under Grants No. #2221943 and #2040961.

References

[1] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9690–9698, 2020. 1

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 2

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *CVPR*, pages 6077–6086, 2018. 6

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1, 2

[5] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. 5

[6] Paola Cascante-Bonilla, Arshdeep Sekhon, Yanjun Qi, and Vicente Ordonez. Evolving image compositions for feature representation learning. In *British Machine Vision Conference (BMVC)*, November 2021. 1

[7] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 5

[8] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, pages 10800–10809, 2020. 2

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. 2, 6

[10] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3164–3174, 2020. 1

[11] Jiafei Duan, Samson Yu, Tan Hui Li, Huaiyu Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *ArXiv*, abs/2103.04918, 2021. 2

[12] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, pages 4495–4504, 2021. 2

[13] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[14] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Damian Mrowca, Michael Lingelbach, Aidan Curtis, Kevin T. Feigelis, Daniel Bear, Dan Gutfreund, David Cox, James J. DiCarlo, Josh H. McDermott, Joshua B. Tenenbaum, and Daniel L. K. Yamins. Threedworld: A platform for interactive multi-modal physical simulation. *ArXiv*, abs/2007.04954, 2020. 1, 3

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, 07–09 Jul 2015. 6

[16] Alberto Garcia-Garcia, Pablo Martinez-Gonzalez, Sergiu Oprea, John Alejandro Castro-Vargas, Sergio Orts, José García Rodríguez, and Alvaro Jover-Alvarez. The robotrix: An extremely photorealistic and very-large-scale indoor dataset of sequences with robot trajectories and interactions. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6790–6797, 2018. 2

[17] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. *arXiv preprint arXiv:2009.08566*, 2020. 2

[18] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 1, 2, 4, 5

[19] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. 7

[20] Vipul Gupta, Zhuowan Li, Chenyu Zhang, Adam Kortylewski, Yingwei Li, and Alan Yuille. Swapmix: Diagnosing and regularizing the over-reliance on visual context in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. 6

[22] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 2

[23] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *ECCV*, pages 53–69, 2018. 2

[24] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, pages 804–813, 2017. 2

[25] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik G. Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *CVPR*, pages 10264–10273, 2020. 6

[26] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997, 2017. 1, 2

[27] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 3

[28] Matthew Johnson-Roberson, Charlie Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *IEEE International Conference on Robotics and Automation (ICRA)*, pages 746–753, 2017. 2

[29] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 198–202, 2017. 1

[30] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *ArXiv*, abs/1712.05474, 2017. 2

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 5

[32] Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1578–1588, 2021. 2

[33] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013. 5

[34] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020. 2, 6

[35] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[36] Pablo Martinez-Gonzalez, Sergiu Oprea, Alberto Garcia-Garcia, Alvaro Jover-Alvarez, Sergio Orts-Escolano, and José A. García-Rodríguez. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 24:271–288, 2019. 2

[37] Enrico Meloni, Luca Pasqualini, Matteo Tiezzi, Marco Gori, Stefano Melacci Dept. of Information Engineering, Mathematics, University of Siena, Dept. of Chemical Engineering, University of Florence, and Maasai Universite Cote dAzur. Sailenv: Learning in virtual visual environments made simple. *25th International Conference on Pattern Recognition (ICPR)*, pages 8906–8913, 2021. 2

[38] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6

[39] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019. 1

[40] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. *Proceedings of the 25th ACM international conference on Multimedia*, 2017. 2

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[42] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *Advances in neural information processing systems*, 28:2953–2961, 2015. 2

[43] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 6

[44] Stephan R Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing photorealism enhancement. *arXiv preprint arXiv:2105.04619*, 2021. 1

[45] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *ArXiv*, abs/2011.02523, 2020. 1, 3, 5

[46] Adrian Lopez Rodriguez and Krystian Mikolajczyk. Domain adaptation for object detection via style consistency. *British Machine Vision Conference (BMVC)*, 2019. 2

[47] Kuniaki Saito, Kohei Watanabe, Y. Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 7

[48] Alonso J. Cerpa Salas, Graciela Meza-Lovon, Manuel Eduardo Loaiza Fernandez, and Alberto Barbosa Raposo. Training with synthetic images for object detection and segmentation in real machinery images. *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 226–233, 2020. 2

[49] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 1

[50] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia

Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 1

[51] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *ICCV*, pages 9338–9346, 2019. 2

[52] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *ArXiv*, abs/2107.06383, 2021. 6

[53] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017. 1

[54] H. Sowizral. Scene graphs in the new millennium. *IEEE Computer Graphics and Applications*, 20(1):56–57, 2000. 3

[55] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Xuan Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. *ArXiv*, abs/2106.14405, 2021. 2

[56] Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. Semantic equivalent adversarial data augmentation for visual question answering. In *European Conference on Computer Vision*, pages 437–453. Springer, 2020. 1

[57] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation, 2018. 2

[58] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 2

[59] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *ArXiv*, abs/1412.3474, 2014. 7

[60] UE4Arch. Ue4arch, 2021. 2

[61] Zeyu Wang, Klint Qinami, Yannis Karakozis, Kyle Genova, Prem Qu Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, pages 8916–8925, 2020. 7

[62] Spencer Whitehead, Hui Wu, Yi Ren Fung, Heng Ji, Rogerio Feris, and Kate Saenko. Learning from lexical perturbations for consistent visual question answering. *arXiv preprint arXiv:2011.13406*, 2020. 2

[63] Spencer Whitehead, Hui Wu, Heng Ji, Rogério Schmidt Feris, Kate Saenko, and Uiuc MIT-IBM. Separating skills and concepts for novel visual question answering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5628–5637, 2021. 5

[64] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. *ArXiv*, abs/1910.01442, 2020. 1, 2

[65] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, pages 6274–6283, 2019. 6

[66] Riccardo Zanella, Alessio Caporali, Kalyan Tadaka, Daniele De Gregorio, and Gianluca Palli. Auto-generated wires dataset for semantic segmentation with domain-independence. *2021 International Conference on Computer, Control and Robotics (ICCCR)*, pages 292–298, 2021. 2

[67] Mingda Zhang, Tristan D. Maidment, Ahmad Diab, Adriana Kovashka, and Rebecca Hwa. Domain-robust vqa with diverse datasets and methods but no target labels. In *CVPR*, pages 7042–7052, 2021. 1