Visual Question Answering on 360° Images

Shih-Han Chou[1,2], Wei-Lun Chao[3], Wei-Sheng Lai[5], Min Sun[2], Ming-Hsuan Yang[4,5]

[1]University of British Columbia [2]National Tsing Hua University
[3]The Ohio State University [4]University of California at Merced [5]Google

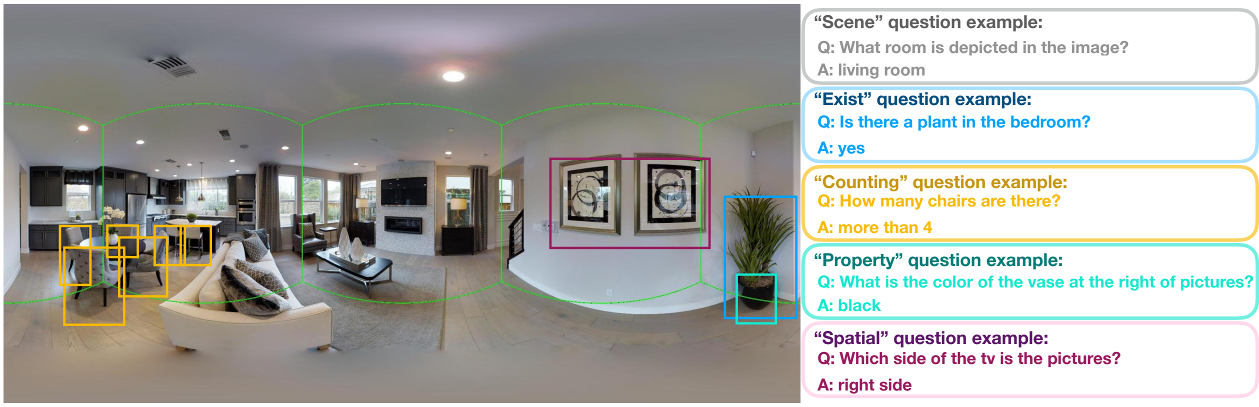
"Scene" question example:
Q: What room is depicted in the image?
A: living room
"Exist" question example:
Q: Is there a plant in the bedroom?
A: yes
"Counting" question example:
Q: How many chairs are there?
A: more than 4
"Property" question example:
Q: What is the color of the vase at the right of pictures?
A: black
"Spatial" question example:
Q: Which side of the tv is the pictures?
A: right side

Figure 1: **An example of our VQA 360° dataset.** We introduce VQA 360°, a novel task of visual question answering on 360° images, and collect the first real VQA 360° dataset, in which each image is annotated with around 11 questions of five types (marked by different colors). The bounding boxes indicate where to look to infer the answers. Best viewed in color.

Abstract

In this work, we introduce VQA 360°, a novel task of visual question answering on 360° images. Unlike a normal field-of-view image, a 360° image captures the entire visual content around the optical center of a camera, demanding more sophisticated spatial understanding and reasoning. To address this problem, we collect the first VQA 360° dataset, containing around 17,000 real-world image-question-answer triplets for a variety of question types. We then study two different VQA models on VQA 360°, including one conventional model that takes an equirectangular image (with intrinsic distortion) as input and one dedicated model that first projects a 360° image onto cubemaps and subsequently aggregates the information from multiple spatial resolutions. We demonstrate that the cubemap-based model with multi-level fusion and attention diffusion performs favorably against other variants and the equirectangular-based models. Nevertheless, the gap between the humans' and machines' performance reveals the need for more advanced VQA 360° algorithms. We, therefore, expect our dataset and studies to serve as the benchmark for future development in this challenging task. Dataset, code, and pre-trained models are available online.[1]

1. Introduction

Visual question answering (VQA) has attracted significant attention recently across multiple research communities. In this task, a machine needs to visually perceive the environment, understand human languages, and perform multimodal reasoning—all of them are essential components to develop modern AI systems. Merely in the past three years, more than two dozen datasets have been published, covering a wide variety of scenes, language styles, as well as reasoning difficulties [2, 17, 19, 23, 35, 36, 48]. Together with those datasets are over a hundred algorithms being developed, consistently shrinking the gap between humans' and machines' performance [4, 16, 24, 25, 26].

Despite such an explosive effort, existing work is constrained in the way a machine visually perceives the world. Specifically, nearly all the datasets use normal field-of-view (NFOV) images taken by consumer cameras. Convolutional neural networks (CNNs) that are carefully designed for such images [21, 37] have been necessary to extract powerful visual features. Nevertheless, NFOV images are not the only way, and very likely not the most efficient way, for a machine to interact with the world. For example, considering a 360° horizontally surrounding scene, the NFOV of a consumer camera can only capture an 18% portion [42]. Such a fact, together with the reduced price of 360° cameras (e.g.,

[1] http://aliensunmin.github.io/project/360-VQA/

Ricoh Theta S, Samsung Gear 360, and GoPro Omni), has motivated researchers to dig into 360° vision [9, 10, 22, 40]. We could imagine every robot to be equipped with a 360° camera in the near future. It is thus desirable to extend VQA to such an informative visual domain.

In this work, we make the first attempt toward VQA on 360° images (VQA 360°). Two major challenges immediately emerge. First, modern deep learning algorithms are heavily data consuming, yet so far, there is no publicly available dataset for VQA 360°. Second, 360° (i.e., equirectangular) images have intrinsic distortion and larger spatial coverage, requiring a novel way to process visual inputs and perform sophisticated spatial reasoning. Specifically, a machine needs to understand the spatial information in questions, search answers across the entire 360° scene, and finally aggregate the information to answer.

To resolve the first challenge, we collect the first real VQA 360° dataset, using 360° images from real-world scenes. Our dataset contains about 17,000 image-question-answer triplets with human-annotated answers (see an example in Figure 1). We have carefully taken the bias issue [19, 24], which many existing VQA datasets suffer, into account in designing our dataset. We thus expect our dataset to benefit the development of this novel task.

In addition, we study two models to address VQA 360°. On the one hand, we use equirectangular images as input, similar to conventional VQA models on NFOV images. On the other hand, to alleviate spatial distortion, we represent an input 360° image by six cubemaps [20]. Each map has its own spatial location and suffers less distortion (cf. Figure 2). We develop a multi-level attention mechanism with spatial indexing to aggregate information from each cubemap while performing reasoning. In this way, a machine can infer answers at multiple spatial resolutions and locations, effectively addressing the algorithmic challenge of VQA 360°. Moreover, cubemap-based architecture is flexible to take existing (pre-trained) VQA models as backbone feature extractors on cubemaps, effectively fusing multimodal information and overcoming the limited data issue.

We conduct extensive empirical studies to evaluate multiple variants of these models. The superior performance by the cubemap-based model demonstrates the need to explicitly consider intrinsic properties of VQA 360°, both visually and semantically. By analyzing the gap between the machine's and the human's performance, we further suggest future directions to improve algorithms for VQA 360°.

Our contributions in this work are two-fold:

- We define a novel task named VQA 360°. We point out the intrinsic difficulties compared to VQA on NFOV images. We further collect the first real VQA 360° dataset, which is designed to include complicated questions specifically for 360° images.
- We comprehensively evaluate two kinds of VQA models for VQA 360°, including one that can effectively handle spatial distortion while performing multi-level spatial reasoning. We then point out future directions for algorithm design for VQA 360°.

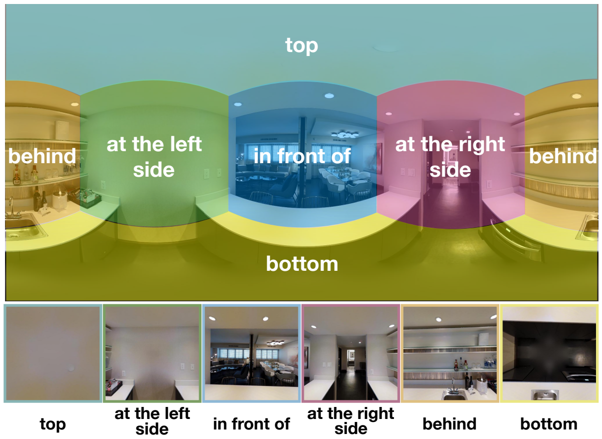

Figure 2: **360° image and cubemaps.** A equirectangular 360° image can be represented by six cubemaps, each corresponding to a spatial location, to reduce spatial distortion.

2. Related Work

VQA models. Visual Question Answering requires comprehending and reasoning with visual (image) and textual (question) information [47]. The mainstream of model architectures is to first learn the joint image-question representation and then predict the answer through multi-way classification. In the first stage, two mechanisms, *visual attention* [1, 44, 34] and *multimodal fusion* [16, 4], have been widely explored. For example, the stacked attention networks (SANs) [45] was developed to perform multi-round attention for higher-level visual understanding. On the other hand, Fukui et al. [16] proposed the Multimodal Compact Bilinear pooling (MCB) to learn a joint representation, and Ben et al. [4] developed a tensor-based Tucker decomposition to efficiently parameterize the bilinear interaction. Recently, several work [8, 32, 33, 39] extended BERT [15] by developing new pre-training tasks to learn (bidirectional) transformers [43] for joint image and text representations.

Despite the variety of architectures, most of existing methods directly apply CNNs to the whole NFOV image to extract (local) features, which may not be suitable to 360° images. In this paper, we explore a different architecture to extract CNN features from the cubemap representations of a 360° image and then fuse features across cubemaps. The cubemap-based model shares some similarity to [1, 45], yet we apply multiple-rounds of attentions to different spatial resolutions, one within and one across cubemaps, so as to achieve better spatial understanding.

VQA datasets. There have been over two dozen of VQA datasets on NFOV images published in recent years. Most of them aim for open-ended answering [2, 19, 30], providing for a pair of image and question with one or multiple correct answers [6, 48]. An alternative setting is multiple-choice answering: a set of candidate answers are provided for each question, in which one of them is correct. Our VQA 360° dataset belongs to the first category but focuses on a very different input domain, 360° images.

We note that there are two emerging VQA tasks, embodied QA [13] and interactive QA [18], that require a machine to interact with the 3D environment (e.g., turn right or move closer). Our dataset and task are different, from two aspects. First, we work on real-world scenes, while both of them are on synthetic ones. Second, we take 360° images as input while they take NFOV images. A machine there has to take actions to explore the environment, being less efficient.

360° vision. With the growing popularity of virtual reality (VR) and augmented reality (AR), 360° images and videos have attracted increasing attention lately. One of the interesting problems is to automatically navigate a 360° video [22, 40, 42] or create a fast-forward summary [31]. Other research topics include 360° video stabilization [29], compression [41], saliency prediction [9], depth estimation [14], and object detection [11, 40]. Recently, Chou et al. [10] study visual grounding to localize objects in a 360° video for a given narrative, while Chen et al. [7] explore natural language navigation in 360° street environments. In contrast to these tasks, VQA on 360° images requires further inferring the answers according to questions, demanding more sophisticated reasoning of the scene.

3. VQA 360° Dataset

We first present the proposed VQA 360° dataset to give a clear look at the task and its intrinsic challenges. We begin with the dataset construction, including image collection, question generation, and answer annotation. We then provide detailed statistics for our VQA 360° dataset.

3.1. Images Collection

We focus on indoor scenes as they are usually more dense with contents such as objects, which are suitable for developing algorithms for sophisticated reasoning. In contrast, outdoor scenes, like those in [22, 31, 41, 42], capture certain (ego-centric) activities and are of sparse contents, which are more suitable for summarization or navigation.

We collect 360° images of indoor scenes from two publicly accessible datasets, Stanford 2D-3D [3] and Matterport3D [5]. Both datasets provide useful side information such as scene types and semantic segmentation, which benefit question generation. There are about 23 different scenes, including common areas in houses (e.g., bathroom, kitchen, bedroom, etc.) and workplaces (e.g., office, conference room, auditorium, etc.). To maximize the image diversity, we discard images captured in the same room but with different viewpoints. In total, we collect 744 images from the Stanford 2D-3D dataset and 746 images from the Matterport3D dataset.

All the 360° images are stored in the equirectangular format and resized to 1024×512. The equirectangular projection maps latitude and longitude of a sphere to the horizontal and vertical lines (e.g., a point at the top of the sphere is mapped to a straight line in an equirectangular image), which inevitably introduces heavy spatial distortion.

3.2. Question Generation

We design several question templates (c.f. Table 1), together with the semantic segmentation and scene types associated with each 360° image[2], to automatically generate questions. Our templates contain five different types: "scene", "exist", "counting", "property" and "spatial". While imposing templates limit the diversity of questions, the main purpose of our dataset is to promote VQA on a new visual domain that has larger spatial coverage and complexity. As illustrated in Figure 1, a 360° image can easily contain multiple objects distributed at multiple locations. *We thus specifically design the question templates—either include spatial specifications or ask for spatial reasoning—to disambiguate the questions and encourage machines to acquire better spatial understanding.* For instance, to answer "What is the color of the vase at the right of pictures?" in Figure 1, a machine needs to first find the pictures (rightmost), look to the right to find the vase, and return the color[3]. To answer "Which side of the TV is the pictures?", a machine needs to detect the TV and picture, and then return their relative spatial information in the scene. Both examples require visual and spatial understanding at multiple resolutions and locations, which are scarce in existing VQA datasets on NFOV images (see the supplementary material for details). On average, we create 11 questions per image.

3.3. Answer Annotations & Question Refinements

We resort to human annotators to provide precise answers. We ask 20 in-house annotators to answer the questions in our dataset. To avoid synonyms words and to ease the process, we offer candidate answers according to the question types for annotators to select directly. Annotators can also type free-form answers if none of the candidates is applicable. We note that the automatically generated questions might be irrelevant to the image or lead to ambiguous answers[4]. In such cases, we instruct the annota-

[2] We can obtain room types and objects appearing in the scenes.

[3] There are three vases in Figure 1. Adding spatial specifications is thus necessary, and different specifications will lead to different answers.

[4] For instance, if there are two chairs with different colors, a question "What is the color of the chair?" will lead to ambiguous answers.

Q type	Template	Example	Answer
Scene	What room is depicted in the image?	What room is depicted in the image?	bedroom/...
Exist	Is/Are there (a) <obj1> ___? + in the <scene> + <direc> + <direc> of the <obj2> + <direc> of the <obj2> in the <scene>	Is there a chair in the kitchen? Is there a chair at my right side? Is there a chair at the right side of the window? Is there a chair at the right side of the window in the kitchen?	yes/no
Counting	How many <obj1> are ___? + in the <scene> + <direc> + <direc> of the <obj2> + <direc> of the <obj2> in the <scene>	How many chairs are in the kitchen? How many chairs are at my right side? How many chairs are at the right side of the window? How many chairs are at the right side of the window in the kitchen?	0/1/2/...
Property	What is the (<color>) <obj1> ___ made of? What is the color of the <obj1> ___? + in the <scene> + <direc> + <direc> of the <obj2> + <direc> of the <obj2> in the <scene>	What is the red sofa in the bedroom made of? What is the red sofa at my right side made of? What is the color of the sofa at the right of the window? What is the color of the sofa at the right of the window in the bedroom?	plastic/wood/... red/brown/...
Spatial	Where can I find the ___<obj1>? Which side of the ___ <obj1> is the ___ <obj2>? + <color> + <material>	Where can I find the white flowers? Which side if the white chair is the wooden door?	in front of you/... right side/...

Table 1: **Question templates and examples.** We design the following question templates and utilize the scene types and semantic segmentation of the images to automatically generate questions.

	Training	Validation	Test
#images	743	148	599
QA pairs	8227	1756	6962
#unique answers	51	51	53
#Scene type Q	765	150	614
#Counting type Q	1986	495	1934
#Existed type Q	2015	417	1655
#Property type Q	1355	322	1246
#Spatial type Q	2106	372	1513

Table 2: **Summary of 360° VQA dataset.** We summarize the number of images, QA pairs, and unique answers in each split of our dataset. We also provide a detailed statistic for each type of question.

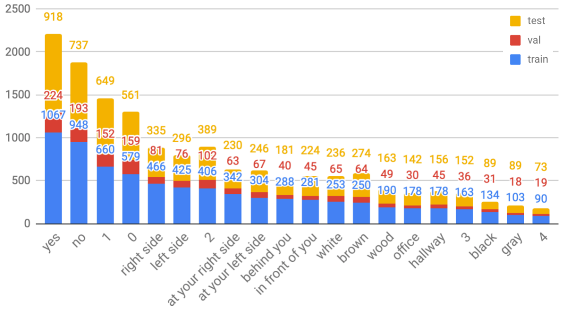

Figure 3: **Distribution of answers.** We balance our dataset such that the answers of the same question type appear uniformly (e.g., “yes/no”, “0/1”, and “right side/left side”).

tors to slightly modify the questions—e.g., by adding spatial specifications—to make them image-related or identifiable. We also instruct annotators to draw bounding boxes (for a subset of image-question pairs), which indicate specific objects or locations associated with the answer. Such information facilitates the analysis of model performances.

3.4. Dataset Statistics

Our VQA 360° dataset consists of $1,490$ images and $16,945$ question-answer pairs, which are split into the training, validation, and test sets with 50%, 10%, and 40% of images, respectively. We summarize the statistics in Table 2 and show the distribution of the top 20 answers in Figure 3. We note that each question type has at least 2 corresponding answers in the top 20 ones. Moreover, those from the same type have the similar number of presence (e.g., “yes/no”, “0/1”, “right/left side”), preventing a machine from cheating by predicting the dominant answer. For question types with a few unique answers, we make sure that the unique answers appear almost uniformly to minimize dataset bias.

4. VQA 360° Models

In this section, we study two VQA models, including one dedicated to resolving inherent challenges in VQA 360°.

Notations and problem definitions. Given a question q and an image i, a machine needs to generate the answer a. One common VQA model is to first extract visual features $f_i = \mathcal{F}_I(i)$ and question features $f_q = \mathcal{F}_Q(q)$, followed

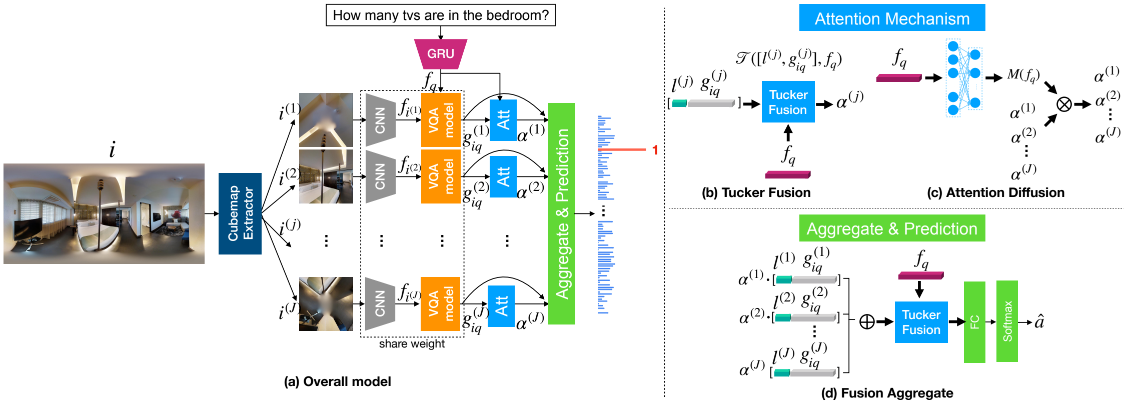

Figure 4: **VQA 360° models.** We propose a cubemap-based architecture that first extracts visual features from the cubemaps of the input 360° image and then performs bottom-up multi-level attention and feature aggregation.

by multimodal representations $g_{iq} = \mathcal{G}(f_i, f_q)$. The multimodal representations are then inputted into a classifier $\mathcal{C}(\cdot)$ of K classes, corresponding to the top K frequent answers, to generate the answer a. Representative choices for $\mathcal{F}_I(\cdot)$ and $\mathcal{F}_Q(\cdot)$ are CNN and RNN models [45], respectively.

4.1. Equirectangular-based Models

As the most common format to store and display a 360° image is the equirectangular projection into a 2D array, we can indeed directly apply existing (pre-trained) VQA models for VQA 360°. We take the Multimodal Low-rank Bilinear Attention Network (MLB) model [26] as an example, which adopts an efficient bilinear interaction for $\mathcal{G}(f_i, f_q)$. We first extract the visual features f_i by the pre-trained ResNet-152 [21] and adopt the Gated Recurrent Units (GRU) [12, 28] to extract the question features f_q. We then input the resulting $g_{iq} = \mathcal{G}(f_i, f_q)$ into a fully-connected layer with K output units to build a K-way classifier $\mathcal{C}(\cdot)$. We optimize the whole network using the training set of our VQA 360° dataset and set K to be the number of unique training answers (i.e., 51).

The MLB model $\mathcal{G}(f_i, f_q)$ pre-trained on the VQA-1 [2] dataset requires f_i to retain a 14×14 spatial resolution, equivalent to inputting a 448×448 image to the ResNet. We thus adopt a few strategies, including cropping or resizing the original 360° image, or inputting the original image while resizing the output ResNet features into a 14×14 spatial resolution by an average pooling layer. We analyze these strategies in Section 5.

Challenges. While the above strategies allow us to exploit VQA models pre-trained on much larger NFOV datasets (e.g., VQA-1 [2]), applying CNNs directly on 360° images suffers the inherent spatial distortion [40]. On the other hand, adopting specifically designed spherical convolutions [40] prevents us from leveraging existing models and pre-trained weights. An intermediate solution that takes both concerns into account is thus desirable.

Moreover, existing VQA models like MLB [26] and SAN [45] only consider a single visual resolution when performing feature aggregation in $\mathcal{G}(f_i, f_q)$. For 360° images that cover a large spatial range, a more sophisticated mechanism that involves multiple resolutions of feature aggregation is required. To this end, we propose a cubemap-based model to simultaneously tackle the above challenges.

4.2. Cubemap-based Models

To reduce spatial distortion, we first represent a 360° image by six non-overlapping cubemaps, $\{i^{(j)}\}_{j=1}^{J}$, via the perspective projection (c.f. Figure 2; see the supplementary material for details). Each cubemap corresponds to a specific portion of the 360° image with less distortion. Collectively, the cubemaps together can recover the original image. This representation naturally leads to a bottom-up architecture that begins with the local region understanding and then global reasoning (cf. Figure 4).

In the first stage, we can apply any existing VQA models, e.g., MLB [26], to each cubemap individually, resulting in J local multimodal representations:

$$g_{iq}^{(j)} = \mathcal{G}(f_{i^{(j)}}, f_q)\,, \qquad (1)$$

where $f_{i^{(j)}}$ denotes the visual features of the j-th cubemap.

Bottom-up multi-level attention. In the second stage, the main challenge is to effectively aggregate information from cubemaps. While average and max pooling have been widely used, they simply ignore the location associated with each cubemap. We thus resort to the attention mechanism:

$$g_i = \sum_{j=1}^{J} \alpha^{(j)} g_{iq}^{(j)}\,, \quad \text{s.t. } \alpha^{(j)} \geq 0,\ \sum_{j} \alpha^{(j)} = 1. \qquad (2)$$

The attention weight $\alpha^{(j)}$ can be computed according to information of each cubemap, *including its location*, making aggregation more flexible. As many existing VQA models already apply the attention mechanism *within* the input images [26, 45] (e.g., a cubemap in our cases), the attention to aggregate *across* cubemaps is actually the second-level of attention but on a coarse resolution.

We apply Tucker fusion $\mathcal{T}(\cdot, \cdot)$ [4] to compute the attention weights according to the cubemap feature $g_{iq}^{(j)}$, location indicator $l^{(j)}$, and question feature f_q: Tucker fusion has been shown effective and efficient in fusing information from multiple modalities. The resulting $\alpha^{(j)}$ is as follows,

$$\alpha^{(j)} = \text{softmax}\{\mathcal{T}([l^{(j)}, g_{iq}^{(j)}], f_q)\}\,, \tag{3}$$

where $[\cdot, \cdot]$ means concatenation. The softmax is performed over $j \in \{1, \cdots, J\}$. We use a one-hot vector $l^{(j)}$ to encode the cubemap location. In this way, the attention weights can zoom into the cubemap location mentioned in the question.

Attention diffusion. The attention weighs by (3); however, do not explicitly consider spatial relationship across cubemaps. For a question like "Is there a chair at the right side of the window?", we would expect the model to first attend to the cubemap that contain the window, and then *shift* its attention to the cubemap at the right. To incorporate such a capability, we learn a diffusion matrix $M(f_q)$ conditioned on the question f_q: the entry $M(f_q)_{u,v}$ indicates how much attention to be shifted from the cubemap v to u. The resulting formula for g_i in (2) becomes:

$$g_i = \sum_{u=1}^{J}\left(\sum_{v=1}^{J} M(f_q)_{u,v}\alpha^{(v)}\right) g_{iq}^{(u)}, \text{s.t.} \sum_{u=1}^{J} M(f_q)_{u,v} = 1. \tag{4}$$

Answer prediction. The resulting feature g_i in (4) or (2) then undergoes another Tucker fusion to extract higher-level image-question interactions before inputted into the classifier $\mathcal{C}(\cdot)$. We can also replace $g_{iq}^{(j)}$ in (4) or (2) by the concatenation of $g_{iq}^{(j)}$ and $l^{(j)}$ to incorporate location cues into g_i. This strategy is, however, meaningless to average or max pooling—it simply results in an all-one vector. We illustrate the overall model architecture in Figure 4. More details are included in the supplementary material.

5. Experimental Results

5.1. Setup

Variants of cubemap-based models. The cubemap-based model can take any existing VQA model as the backbone. We choose the MLB model [26], a bilinear multimodal fusion and attention model. We experiment with other VQA backbones [4, 38] in the supplementary material to demonstrate the applicability of the cubemap-based models.

We remove the fully-connected layer of the original MLB model to extract multimodal features. We apply the pre-trained MLB model to each cubemap of size 448 × 448, and consider the following three different *aggregation schemes* before performing the final answer prediction.

- CUBEMAP-AVGPOOL: apply average pooling on $g_{iq}^{(j)}$.
- TUCKER: attention weights by Tucker fusion in (3).
- TUCKER&DIFFUSION: attention weights by Tucker fusion followed by the diffusion in (4).

Variants of equirectangular-based models. We consider four ways to apply MLB on the equirectangular images.

- CENTRAL-CROP: resize the shorter size of the image to 448 to preserve the aspect ratio and then crop the image to 448 × 448 to extract ResNet features.
- RESIZE: resize the image into 448 × 448 without any cropping and extract ResNet features.
- RESNET-AVGPOOL: resize the shorter size of the image to 448 and apply an average pooling layer on the ResNet output to obtain 14 × 14 resolution features.
- DIRECT-SPLIT: split an equirectangular image into 2×3 patches, resize each to 448×448 and apply MLB, and then apply TUCKER&DIFFUSION to aggregate information for predicting the answer.

Note that the DIRECT-SPLIT and TUCKER&DIFFUSION models have the same architecture but different inputs.

Baselines. We provide Q-TYPE PRIOR, a model that outputs the most frequent answer of each question type.

Implementation details. We first pre-train the backbone MLB model on the VQA-1 [2] dataset, which contains over 100,000 NFOV images and 300,000 question-answer pairs for training. Then, we plug the pre-trained model in all the compared models and fine-tune the models on our VQA 360° training set for 150 epochs. We optimize our models with the ADAM [27] optimizer and select the model with the best performance on the validation set.

Evaluation metric. We use the top-1 accuracy for evaluation. We report two types of accuracy: the average accuracy i) over all the questions, and ii) over question types.

5.2. Analysis and Discussions

Table 3 summarizes the results on VQA 360° test set. The cubemap-based model with TUCKER&DIFFUSION for attention weights performs favorably against other models, demonstrating the effectiveness of multi-level and *diffused* attention on top of cubemaps representation for VQA 360°. In the following, we discuss several key observations.

Limited language bias. The top row (Q-type prior) in Table 3 examines the dataset bias, which predicts the most frequent answer of each question type. The inferior results suggest a low language bias in our dataset. Specifically, for "exist" type questions that only have two valid answers each (i.e, "yes" or "no"), using language prior is close to random guess. Machines need to rely on images to answer.

Model	Variants	Overall avg	Avg by type	Scene	Exist	Counting	Property	Spatial
Q-TYPE PRIOR	-	33.50	31.71	25.41	55.47	33.56	21.99	22.14
Equirectangular-based	CENTRAL-CROP	53.39	54.07	60.66	75.00	47.10	50.16	37.45
Equirectangular-based	RESIZE	54.21	55.77	68.46	75.66	**47.31**	51.48	35.96
Equirectangular-based	RESNET-AVGPOOL	54.47	56.14	69.34	76.81	46.32	50.96	37.25
Equirectangular-based*	RESNET-AVGPOOL	54.15	55.55	67.48	77.17	46.17	49.04	37.90
Equirectangular-based	DIRECT-SPLIT	54.77	56.59	71.36	75.75	46.68	49.56	39.62
Cubemap-based	CUBEMAP-AVGPOOL	54.60	56.23	69.17	76.22	46.79	**51.72**	37.26
Cubemap-based	TUCKER	57.71	59.07	69.89	**77.23**	46.53	48.24	53.47
Cubemap-based	TUCKER&DIFFUSION	**58.66**	**60.26**	**72.01**	76.34	46.84	50.12	**55.98**
Cubemap-based*	TUCKER&DIFFUSION	54.09	55.54	67.65	76.16	45.91	48.60	39.39

Table 3: **Quantitative results on the VQA 360° test set.** The * models are trained from scratch on the VQA 360° training set without pre-training on the VQA-1. The best result of each column is marked by the bold black color.

Equirectangular-based models. As shown in Table 3, the RESNET-AVGPOOL model outperforms the CENTRAL-CROP and RESIZE, indicating the poor applicability of cropping and resizing to 360° images. Since 360° images have large spatial coverage, in which objects might be of small sizes, resizing will miss those small objects while central cropping will lose 50% of the image content.

Cubemaps v.s. Equirectangular input. One major issue of applying existing VQA models directly to the 360° images is the spatial distortion. This is justified by the fact that all the equirectangular-based models are outperformed by all the cubemap-based models (except the CUBEMAP-AVGPOOL one) on the overall performance. Specifically, by comparing the DIRECT-SPLIT and TUCKER&DIFFUSION, whose main difference is the input, the $3 \sim 4\%$ performance gap clearly reflects the influence of distortion. By looking into different question types, we also observe consistent improvements by applying cubemaps.

Pre-training. Comparing the models with $\star$ (trained from scratch) and without $\star$ (with pre-training), the pre-trained weights (from the VQA-1 dataset) benefits the overall performance, especially for the cubemap-based models.

Attention. Applying cubemaps resolves one challenge of VQA 360°: spatial distortion. We argue that a sophisticated way to aggregate cubemaps features to support spatial reasoning is essential to further boost the performance. This is shown from the improvement by TUCKER&DIFFUSION or TUCKER, compared to CUBEMAP-AVGPOOL: the former two apply attention mechanisms guided by questions and cubemap locations for multi-level attention. Specifically, TUCKER&DIFFUSION outperforms CUBEMAP-AVGPOOL by a notable 3.4% at Avg. by Q type, mostly from the "spatial" question type. TUCKER&DIFFUSION with spatial *diffusion* also outperforms TUCKER in all the question types.

Location feature. Concatenating $l^{(j)}$ with $g_{iq}^{(j)}$ in (2) and (4) enables our model to differentiate cubemaps. Table 4 compares the TUCKER&DIFFUSION and TUCKER with/without $l^{(j)}$. The location indicator leads to consistent improvement, especially on the "spatial" type questions.

Model	Avg.	Avg. by Q type	Spatial
TUCKER (w/o)	53.81	53.81	36.09
TUCKER (w/)	57.71	59.07	53.47
TUCKER&DIFFUSION (w/o)	54.91	56.51	39.13
TUCKER&DIFFUSION (w/)	58.66	60.26	55.98

Table 4: **Comparison of w/ and w/o location feature.**

Model	Overall	Scene	Exist	Counting	Property	Spatial
Human	84.05	88.95	91.79	71.58	89.97	85.25
Machine	59.80	68.89	77.12	49.65	45.81	61.97

Table 5: **Results of human evaluation.** We also include the machine's performance on the same 1,000 questions to analyze the humans' and machines' gap.

Human Evaluation. We conduct a user study on our VQA 360° dataset. We sample $1,000$ image-question-answer triplets from the test set and ask at least two different users to answer each question. To ease the process, we give users five candidate answers, including the correct answer and four other answers that are semantically related to the question. There are a total of 50 unique users participating in the user study. We note that the annotators labeling our dataset are not involved in the human evaluation to avoid any bias.

We summarize the results of human evaluation and the machine's prediction[5] in Table 5. Humans achieve a 84.05% overall accuracy, which is at the same level as many existing VQA datasets [2, 6, 46] and is much higher than another dataset on indoor images [35], justifying the quality of our VQA 360° dataset. Among the five question types, humans perform relatively poorly on "counting", which makes sense due to the complicated contents of 360° images and the possible small objects. Overall, there is about $\sim 25\%$ performance gap between human and machines. The gap is larger especially on "counting", "property", and "spatial" types, suggesting the directions to improve algorithms so as to match humans' inference abilities.

[5] We use our best cubemap-based model TUCKER&DIFFUSION.

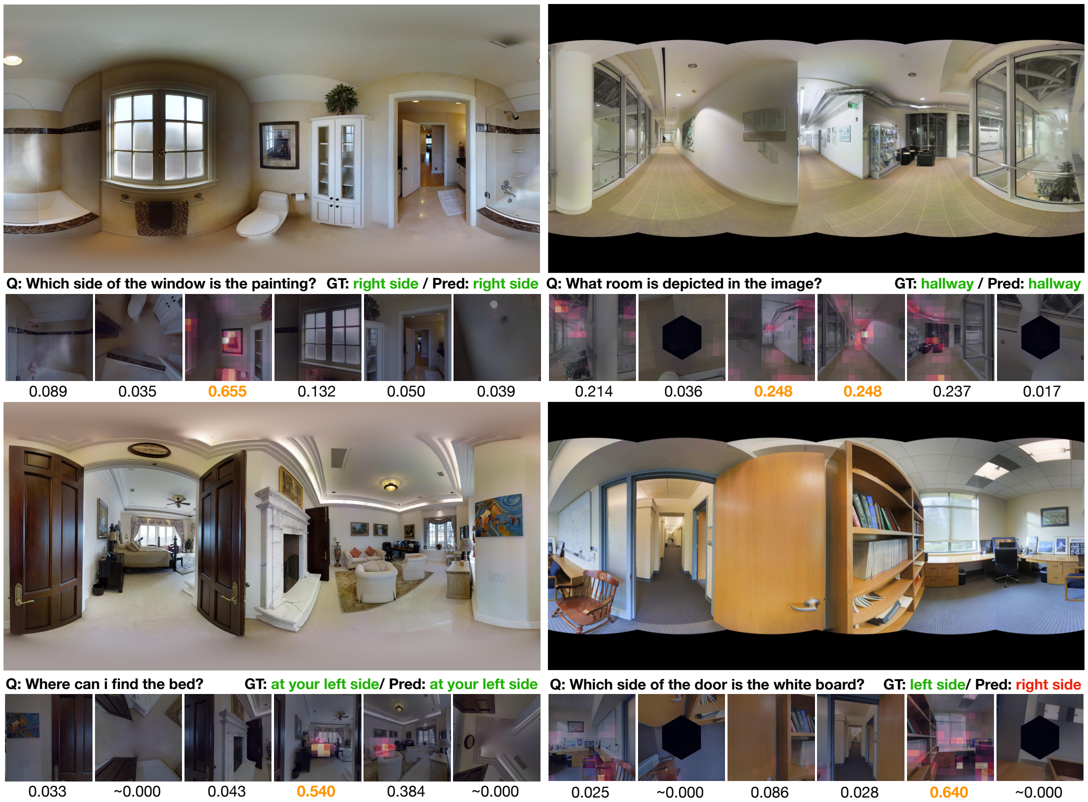
Q: Which side of the window is the painting? GT: right side / Pred: right side
0.089
0.035
0.655
0.132
0.050
0.039
Q: What room is depicted in the image? GT: hallway / Pred: hallway
0.214
0.036
0.248
0.248
0.237
0.017
Q: Where can i find the bed? GT: at your left side/ Pred: at your left side
0.033
~0.000
0.043
0.540
0.384
~0.000
Q: Which side of the door is the white board? GT: left side/ Pred: right side
0.025
~0.000
0.086
0.028
0.640
~0.000

Figure 5: **Visualization of attention.** We use the cubemap-based model TUCKER&DIFFUSION as it performs the best. The digits below the cubemaps indicate the attention across cubemaps. The heat maps indicate the attention within cubemaps.

Qualitative results. We present qualitative results in Figure 5. Besides showing the predicted answers, we visualize the attention weights across cubemaps (by the digits) and within cubemaps (by the heat maps). The cubemap-based model with TUCKER&DIFFUSION can zoom in to the cubemaps related to the questions, capture the answer regions, and aggregate them to predict the final answers. Take the question "Which side of the window is the painting?" for example (the top-left one of Figure 5). The model puts high attention on the cubemaps with windows and pictures and is able to infer the relative location. For the question "What room is depicted in the image?" (the top-right of Figure 5), the model distributes attention to all cubemaps except the top and bottom ones to learn information through them. We also show a failure case in the bottom-right of Figure 5. The question asks "Which side of the door is the whiteboard?". However, the model mistakenly recognizes the window as the white board and incorrectly answers "right side".

6. Discussion and Conclusion

We introduce VQA 360°, a novel VQA task on a challenging visual domain, 360° images. We collect the first VQA 360° dataset and experiment with multiple VQA models. We then present a multi-level attention model to effectively handle spatial distortion (via cubemaps) and perform sophisticated reasoning. Experimental results demonstrate the need to explicitly model intrinsic properties of 360° images, while the noticeable gap between humans' and machines' performance reveals the difficulty of reasoning on 360° images compared to NFOV images.

We surmise that the gap may partially be attributed to the hand-crafted cubemap cropping. On one end, objects appear around the cubemap boundaries may be splitted. On the other end, it requires specifically designed mechanisms (e.g., attention diffusion (4)) to reason the spatial relationship among cubemaps. These issues likely explain the human-machine gap at the "counting" and "spatial" questions. Thus, to advance VQA 360°, we suggest developing image-dependent cropping that detects objectness regions from the equirectangular images. We also suggest developing a back-projection-and-inference mechanism that back-projects the detected objects into the 360° environment and performs reasoning accordingly. Besides, the current questions are generated (or initialized) by templates. A future work is to include more human efforts to increase the question diversity. We expect our dataset and studies to serve as the benchmark for the future developments.

Acknowledgments. This work is supported in part by NSF CAREER (# 1149783) and MOST 108-2634-F-007-006 Joint Research Center for AI Technology and All Vista Healthcare, Taiwan.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 1, 3, 5, 6, 7

[3] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv*, 2017. 3

[4] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017. 1, 2, 6

[5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 3

[6] W.-L. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 3, 7

[7] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, 2019. 3

[8] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 2

[9] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *CVPR*, 2018. 2, 3

[10] S.-H. Chou, Y.-C. Chen, K.-H. Zeng, H.-N. Hu, J. Fu, and M. Sun. Self-view grounding given a narrated 360° Video. In *AAAI*, 2018. 2, 3

[11] S.-H. Chou, C. Sun, W.-Y. Chang, W.-T. Hsu, M. Sun, and J. Fu. 360-indoor: Towards learning real-world objects in 360° indoor equirectangular images. *arXiv*, 2019. 3

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*, 2014. 5

[13] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *CVPR*, 2018. 3

[14] G. P. de La Garanderie and A. Atapour. Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360 panoramic imagery. In *ECCV*, 2018. 3

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018. 2

[16] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv*, 2016. 1, 2

[17] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *NIPS*, 2015. 1

[18] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. IQA: Visual question answering in interactive environments. In *CVPR*, 2018. 3

[19] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 1, 2, 3

[20] N. Greene. Environment mapping and other applications of world projections. *IEEE CGA*, 1986. 2

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[22] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In *CVPR*, 2017. 2, 3

[23] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 1

[24] K. Kafle and C. Kanan. An analysis of visual question answering algorithms. In *ICCV*, 2017. 1, 2

[25] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv*, 2017. 1

[26] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017. 1, 5, 6

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[28] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 5

[29] J. Kopf. 360° video stabilization. *ACM TOG*, 2016. 3

[30] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and F.-F. Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 3

[31] W.-S. Lai, Y. Huang, N. Joshi, C. Buehler, M.-H. Yang, and S. B. Kang. Semantic-driven generation of hyperlapse from 360° video. *TVCG*, 2017. 3

[32] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[33] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2

[34] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 2

[35] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 1, 7

[36] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 1

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[38] A. Singh, V. Goswami, V. Natarajan, Y. Jiang, X. Chen, M. Shah, M. Rohrbach, D. Batra, and D. Parikh. Pythia-a platform for vision & language research. In *SysML Workshop, NeurIPS*, volume 2018, 2018. 6

[39] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[40] Y.-C. Su and K. Grauman. Learning spherical convolution for fast features from 360° imagery. In *NIPS*, 2017. 2, 3, 5

[41] Y.-C. Su and K. Grauman. Learning compressible 360° Video video isomers. In *CVPR*, 2018. 3

[42] Y.-C. Su, D. Jayaraman, and K. Grauman. Pano2Vid: Automatic cinematography for watching 360° videos. In *ACCV*, 2016. 1, 3

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[44] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[45] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 2, 5, 6

[46] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In *ICCV*, 2015. 7

[47] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun. Leveraging video descriptions to learn video question answering. In *AAAI*, 2017. 2

[48] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 3