

Point and Ask: Incorporating Pointing into Visual Question Answering

Arjun Mani, Nobline Yoo, Will Hinthorn*, Olga Russakovsky
 Princeton University
 {arjuns, olgarus}@cs.princeton.edu

Abstract

Visual Question Answering (VQA) has become one of the key benchmarks of visual recognition progress. Multiple VQA extensions have been explored to better simulate real-world settings: different question formulations, changing training and test distributions, conversational consistency in dialogues, and explanation-based answering. In this work, we further expand this space by considering visual questions that include a spatial point of reference. Pointing is a nearly universal gesture among humans, and real-world VQA is likely to involve a gesture towards the target region.

Concretely, we (1) introduce and motivate point-input questions as an extension of VQA, (2) define three novel classes of questions within this space, and (3) for each class, introduce both a benchmark dataset and a series of model designs to handle its unique challenges. There are two key distinctions from prior work. First, we explicitly design the benchmarks to require the point input, i.e., we ensure that the visual question cannot be answered accurately without the spatial reference. Second, we explicitly explore the more realistic point spatial input rather than the standard but unnatural bounding box input. Through our exploration we uncover and address several visual recognition challenges, including the ability to reason both locally and globally about the image, and to effectively combine visual, language and spatial inputs. Code is available at: github.com/princetonvisualai/pointingqa.

1. Introduction

Visual Question Answering (VQA) has emerged as a popular and challenging task in computer vision [3, 10, 31, 13, 7, 6]. When the task was first introduced, the challenge of answering a natural language question about an image was already a significant leap beyond more conventional tasks such as object recognition. Since then, questions in VQA benchmarks have been increasingly growing in complexity: for example, “Are there any cups to the left of the

*WH is currently at Microsoft. Work was conducted while he was an undergraduate student at Princeton University.

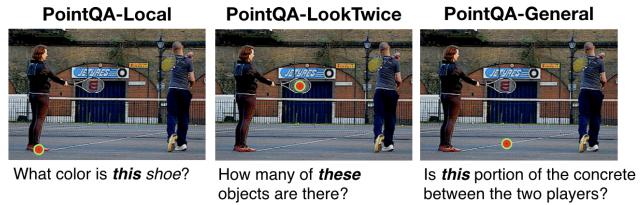


Figure 1: Three types of visual questions requiring a point input (red) which we propose and analyze in this work.

tray on top of the table?” is an example question from the recent GQA [13] benchmark. Such questions effectively test the ability of VQA agents to parse very complex sentences, but arguably are becoming less realistic. In a real-world setting, it’s unlikely that a human would use the phrase “to the left of the tray on top of the table.” It’s far more likely that they would instead ask “Are there any cups over there” and point to the left of the tray. In fact, human psychology literature shows that pointing to interesting objects or situations is one of the first ways by which babies communicate intention [21, 19]. Understanding pointing as part of a visually grounded dialog with humans would naturally be a key ability of real-world AI systems.

We thus propose to expand the space of VQA by considering visual questions that further include a spatial point of reference for context. Prior works have used visual grounding to expand the question space of VQA: e.g., Visual7W [31] introduced “which” question with image regions as candidate answers; Visual Genome [16] contains questions that are associated with particular regions in the image; GQA [13] leverages the grounded scene graph in its question construction process. There are two key distinctions of our proposal from this line of work. First, we explicitly design the benchmarks to *require* the point input, i.e., we ensure that the visual question cannot be answered accurately without the spatial reference. Second, we explore the more realistic *point* spatial input rather than the standard but unnatural bounding box used in [31, 16, 13].

We introduce a set of tasks exploring different aspects and challenges of point-based spatial disambiguation,

shown in Fig. 1. In all tasks the input is an image and a single pixel in the image corresponding to the spatial grounding. The target output is a multiple choice answer.

We first consider two narrow settings: (1) **PointQA-Local**, where only the local region around the point is relevant to the question, e.g., “What color is *this shirt*?” where *this* is specified by a pixel in the image (Sec. 3) and (2) **PointQA-LookTwice**, which requires a global understanding of the image, e.g., “is another shirt *this color*?” (Sec. 4). For each we construct a corresponding dataset from Visual Genome annotations [16], with 57,628 questions across 18,830 images and 57,405 questions across 34,676 images, respectively. We then modify and benchmark VQA models [29, 15] to incorporate the point input.

Finally, we consider the general setting of unconstrained questions which require point disambiguation in **PointQA-General** by adapting human-written questions from the Visual7W [31] dataset, resulting in 319,300 questions over 25,420 images (Sec. 5). We modify the state-of-the-art Pythia [29], MCAN [28] and LXMERT [27] models to incorporate the point input, and demonstrate their effectiveness in this new setting.

To summarize, our work advances VQA along a new dimension. Concretely, we (1) introduce and motivate point-input questions as an extension of VQA, (2) design a set of benchmark datasets, and (3) introduce effective model extensions to handle the unique challenges of this space.

2. Related Work

Spatial Grounding in VQA. Visual grounding has become a central idea in the VQA community [31, 10, 1, 13]. It is increasingly seen as important that VQA models localize the object being asked about to answer a visual question. This idea has influenced the development of several datasets, including Visual7W [31] and GQA [13]. In these datasets, bounding boxes are provided for each object mentioned in the question or answer. To encourage grounding and counteract language priors, [10] introduced the VQA 2.0 Dataset in 2017, which consists of complementary images for each question. [1] further introduce the VQA-CP dataset, where priors differ in training and test splits. All these works indicate the importance of visual grounding for the VQA task.

A few works have used visual grounding to expand the question space of VQA. The authors of Visual7W introduce a *pointing QA* task, involving a ‘which’ question and image regions as candidate answers [31]. They use their manually collected object-level groundings to construct such questions. A number of works have also explored grounding in the context of embodied question answering, which requires an agent to explore an environment to answer a visual question [6, 9]. Other grounding-based questions include region-based QAs in Visual Genome [16], where the

question is associated with a particular region of the image. However, it is not necessary for answering the question that the region be provided: we sampled 100 questions randomly and only 17% actually required the region to produce the correct answer. To our knowledge, we are the first to introduce a benchmark where a spatial grounding signal is explicitly required to answer a question.

The importance of visual grounding has also influenced the development of VQA models. In particular, most state-of-the-art VQA models have an attention mechanism on the image [27, 18, 29, 28]. The relative success of these models indicates the importance of successful visual grounding in the VQA task. Several works include pixel-wise prediction as a primary or auxiliary task of the VQA model. [30] mine ground-truth attention maps from Visual Genome and include attention prediction explicitly as an auxiliary task of the model. Other works output a visual justification for the answer as a heatmap [23] or a semantic segmentation of the visual entities relevant to the question [8]. However, the challenge of actually accepting a spatial grounding input into VQA models has not been previously explored.

Point input. Despite being ubiquitous for humans, pointing as a way of communicating intention has been underexplored in computer vision. Studies in the robotics [12, 26, 24] or human-computer interaction [20] communities have largely been limited to simple, constrained environments. In vision, pointing has been used as a form of cost-effective supervision [4, 14, 22, 5] but never examined in depth beyond cost-accuracy tradeoffs. The use of pointing as a communicative gesture in humans [21, 19] motivates deeper study in a computer vision context.

3. PointQA-Local: reasoning about a region

We now begin our exploration of the space of pointing questions, starting with a simpler **PointQA-Local** setting: questions involves queries about the attributes of a particular object (e.g. “What color is *this car*?”). A local region around the point is completely sufficient to answer the question. In Sec. 3.1 we construct a corresponding dataset using existing annotations from Visual Genome [16]. We design the benchmark such that a point is *required* to answer the question, ensuring that the image as a whole contains multiple possible answers.

PointQA-Local allows us to investigate the capabilities of the model to accept a point input, without additional complexities introduced in Sections 4 and 5. Importantly, in Sec. 3.3 we demonstrate that the model is able to understand the question and the task enough to vary attention around the point as needed (consider “What color is this shirt?” versus “What action is this person doing?”).

3.1. PointQA-Local dataset

Our **PointQA-Local** dataset contains questions that require reasoning over an image region, with the point input disambiguating the correct region. We use three templates:

1. What color is *this* [object class]?
2. What shape is *this* [object class]?
3. What action is *this* [object class] doing?

We generate the dataset from Visual Genome [16], which consists of 108,249 images with each image annotated on average with 35 objects, 26 object attributes, and 21 pairwise relationships between the objects.

Attribute selection. We select the 100 most common attributes from Visual Genome (e.g. red, round, small), which account for 70.1% of annotations. We manually group these attributes into four general categories: color, shape, action, and size. We only use the first three categories, as size is a relative rather than an absolute property of the object.

Question Generation. On every image in Visual Genome, we search for examples that satisfy the constraints of **PointQA-Local**. Concretely, we examine pairs of bounding boxes b_i, b_j annotated with corresponding object classes o_i, o_j and the sets of attributes A_i, A_j . If $o_i = o_j$, so the object classes are the same, we then look for a pair of attributes $a_i \in A_i, a_j \in A_j$ such that $a_i \neq a_j$ and a_i, a_j are of the same category (color, shape or action). We construct a question q of the form “What C is this O ?”, where C is the attribute category and $O = o_j = o_k$ the object class. We then add two examples into our dataset: (1) question q , point to the center of the bounding box b_i , and answer a_i , and (2) question q , point to center of b_j and answer a_j .

We performed several optimizations to ensure quality. Instances where $\text{IoU}(b_i, b_j) \geq t$ were filtered out (we set $t = 0.2$ for high precision). Any object with more than one attribute in the same category was excluded (e.g., a striped shirt annotated as both “red” and “white”) to avoid confusion. Visually similar or identical attributes (e.g., ‘blonde’, ‘yellow’) were collapsed into one attribute to reduce noise.

Statistics. The final dataset consists of 57,628 questions across 18,380 images, with 20 unique answers. Due to the high representation of color attributes in Visual Genome, 97.2% of questions are about color, with the remaining being about shape or action. Although biased in this way, the relative simplicity of color questions allows us to focus on the challenge of incorporating a point input. Although most questions have exactly two unique answers without a disambiguating point, 11.9% of questions have greater than two answers. Fig. 2 shows example questions in the dataset which illustrate the challenges of answering point-input questions, especially when the point can be placed on a non-relevant part of an object (Fig 2a, Fig 2c) or on another smaller, occluding object (Fig 2b).

The images are randomly divided into (1) *train*, 70%,



Figure 2: Examples of questions in the **PointQA-Local** Dataset across color, shape, and action. Point is in red.

with 40,409 questions across 12,867 images; (2) *val*, 10% with 5,724 questions across 1,838 images; (3) *test-dev*, 10%, with 5,673 questions across 1,838 images, and (4) *test-final*, 10%, at 5,910 questions across 1,837 images.

Human accuracy. One final question is whether simulating the pointing as simply the center of the bounding box is a sufficiently understandable spatial disambiguation cue, and whether the dataset as a whole is reliable. To evaluate this, we run a small human study on 100 random questions of *test-final*. Humans agreed with the dataset labels 76% of the time, successfully understanding the reference. Please see Appendix A.2 for details on human evaluation.

3.2. PointQA-Local models

We modify the commonly-used Pythia model [2, 29] to incorporate the point input (x, y) . We consider the more complex transformer-based models in Sec. 5; here the language variations of the **PointQA-Local** dataset are not particularly complex and thus we choose a somewhat simpler and easier-to-analyze model.

Background. Standard VQA systems [2, 29, 28] represent the image using a set of bounding box proposals from an object detection model, typically Faster-RCNN [25]. Concretely: (a) a Region Proposal Network extracts a set of candidate regions along with their “objectness” scores, (b) non-max suppression is performed to remove similar boxes, and (c) the top remaining N regions are processed by an ROI Pool operation to generate a set of visual features $\mathbf{v}_1 \dots \mathbf{v}_N, \mathbf{v}_i \in \mathbb{R}^D$. In step (d), the question is encoded by a recurrent model into $\mathbf{q} \in \mathbb{R}^M$. Then: (e) \mathbf{q} is combined with $\mathbf{v}_1 \dots \mathbf{v}_N$ to compute a normalized attention vector $\mathbf{a} \in \mathbb{R}^N$ over the N region proposals, (f) the probabilistic attention \mathbf{a} is used to compute a weighted average of the region features and obtain the image representation $\mathbf{v} \in \mathbb{R}^D$, and (g) \mathbf{q} and \mathbf{v} are projected to a common vector space, combined via element-wise multiplication followed by fully-connected layers and a softmax activation to predict over the answer distribution. Steps (a)-(c) use a pre-trained detection model, while (d)-(g) are trained.

Modifications. We choose the simplest way of incorporating the point input into the system above: in step (a)



Figure 3: The Pythia-based local region model (Sec. 3.2) successfully varies attention around the point in response to a question. (Darker = higher attention; point in red).

above, we remove all candidate regions that do not contain the point (x, y) . The rest of the pipeline is unchanged, apart from zero-padding to N regions if necessary (we use $N = 100$, as is standard). Our approach produces the effect of a pointing gesture by restricting the view of the model to objects around the point. It has the additional advantage of allowing us to analyze the learned attention around the point, yielding insights into the behavior of the model.

3.3. PointQA-Local evaluation

We benchmark the model extensions on the **PointQA-Local** dataset, focusing on insights into the new task.

Implementation details. We train the model on the *train* subset of the data. For extracting the region features, we use a Feature Pyramid Network (FPN) [17] with a ResNet-101 [11] backbone and IOU of 0.5 for non-maximum suppression, as in the VQA Challenge implementation of Pythia [29]. The model was trained using AdaMax with a learning rate of 0.002. We use early stopping on the *val* set with patience of 500 iterations.

Ablation studies. Table 1 reports results on *test-dev* where we experiment with several different methods for incorporating a point input. Our strategy of removing regions not containing the point and allowing the model to learn attention over the rest achieves at least 5.8% improvement over simpler strategies such as removing all but the smallest or the highest-scoring region proposal containing the point. Improvement is particularly significant on action questions, where other methods perform 12.6% worse.

Attention analysis. An average of 27 region proposals containing the point are provided as input to the model. The attention is relatively peaked, with the maximum attention of 0.548 on average in *test-dev*. An important property is the model’s ability to attend based on the question. For the 799 questions in *test-dev* that ask “What color is this shirt?”, we can change the question to “What action is this person doing?” This increases the median size of the max-attention region from 2,997 to 5,451 pixels, indicating that the model shifts its attention accordingly. (Fig. 3).

	Strategy	QIPBA	Overall	Color	Action	Shape
Priors	Q-only	+ - - -	27.8	25.8	45.5	32.1
	Modal-A	- - - +	52.2	52.0	60.1	52.8
	Full Img	+ + + -	37.4	37.2	45.4	35.9
	Top-score	+ + + -	44.8	44.4	62.2	43.4
	Model Smallest	+ + + -	69.2	69.6	58.0	56.6
	Ours	+ + + -	75.0	75.4	66.4	56.6
	GT box	+ + - -	80.2	80.7	67.1	60.4

Table 1: Accuracy (in %) of different models and baselines on the Local-QA *test-dev* split. The QIPBA column indicates the model input: **Q**uestion, **I**mage, **P**oint, ground truth **B**ounding box, and/or the set of all correct **A**nswers in the image. Modal-A is a baseline oracle that selects the mode answer among all answers that are *correct* in this image. Q-only relies only on dataset language priors; Full Img is the original VQA model, not using the point input; Top-score and Smallest take the features of only the highest-scoring or smallest region proposal containing the point respectively (thus restricting the attention to a single region). Our proposed strategy outperforms these alternatives and even performs close to when the ground truth box instead of the point is provided during training and testing (bottom row).



Figure 4: In both examples, the standard verbal-only VQA model (top) fails to understand the verbal disambiguation and picks an answer corresponding to the wrong object instance; ours (bottom) correctly incorporates the point input.

Spatial vs Verbal Disambiguation. Finally, we note for many **PointQA-Local** questions, a disambiguating phrase could be provided as a substitute for a point (e.g. “What color is the car *on the left*?”) Such verbal disambiguation is often unnatural and can grow expensive depending on the object of interest; this partially explains why humans prefer to point and motivates pointing as a more realistic setup for a broad set of visual questions. We provide a brief comparison between verbal and spatial disambiguation.

Data. To do so, we collect a small dataset using human-written questions from Visual Genome [16]. A parser is used to detect the question subject, and we (1) ensure this object appears multiple times in the image and (2) search for prepositional phrases appended to the question subject to detect verbal disambiguation. Using object/attribute annotations, the question’s subject is matched to an object in the

image, and a “spatial version” of the question is generated by removing the verbal disambiguation and instead generating a point from the center of the target object’s bounding box. We thus obtain a verbally disambiguating version of the dataset D_V and a spatial disambiguating version D_S , with 1,855 questions across 1,575 images in each. We use 80% for training and 10% each for validation and testing.

Results. Our model leveraging the point input achieves an accuracy of 66.5% on D_S , while the Pythia [29] baseline relying on verbal disambiguation achieves only 26.5% on D_V (Fig. 4). Note that a question-only model gets 22.2% accuracy on D_V ; thus, the baseline does not effectively understand verbal disambiguation. This further motivates point-input questions as an extension of VQA.

4. PointQA-LookTwice: reasoning about a local region in the broader image context

We next consider the more general **PointQA-LookTwice** setting which requires situating a local region in the broader context of the image. A natural example is counting questions such as “How many of *this* animal are there?”, where the model must identify the relevant object around the point and then use this information to attend to the entire image. Correspondingly, in Section 4.1 we construct a dataset of counting questions from Visual Genome [16] such that looking at a local region around the point is insufficient to answer the question. We then introduce a new model in Section 4.2 that includes global attention and show that it successfully reasons about the point in the broader image context.

4.1. PointQA-LookTwice dataset

We construct the **PointQA-LookTwice** dataset leveraging the 99,860 human-written counting questions in Visual Genome [16] (e.g., “how many trucks are there?”). We turn it into pointing QA by further leveraging the object annotations as in Sec. 3.1.

Question Generation. For each question, we extract the subject of the question (e.g., “truck”) and attempt to match it to an annotated region in the image. If multiple regions exist, one is chosen at random; if none exist, the question is removed. From this filtered question set, questions about object classes appearing less than 100 times total are further removed. The remaining set of object classes are manually grouped into three super-categories: beings (people and animals), vehicles (ex. cars, planes), and objects (ex. laptops, umbrellas). From each question of the form e.g., “How many [object class] are there?” we generate two new questions by replacing the “[object class]” either with (1) “these [supercategory]” or (2) just “these.” We disambiguate the reference through the point input, simulated as the center of the object bounding box. The answer remains the same for



Figure 5: Example entries in the **PointQA-LookTwice** Dataset. Point is in red and supercategory is underlined.

all three questions; we thus train the model to locally infer the object category from the point and then count globally. From the raw object counts we bin the possible answers to “1”, “2”, and “ > 2 ”, such that the model predicts one of three answers to the counting question.

Counteracting priors. Our goal is to construct questions that *require* the point input and cannot be answered without it. Thus, we enforce that for each image I in the evaluation set, there are at least two questions q_i, q_j such that $o_i \neq o_j$ (different objects), and $a_i \neq a_j$ (different answers).

The remaining generated questions are added to the training set. To prevent priors in the training set, for each question q_i we use Visual Genome object annotations to generate a question q_j with $o_i \neq o_j$ and $a_i \neq a_j$. Object counts are generated from the IoU-filtered object annotations in the image and may be imperfect due to under-counting (thus not used during evaluation). The original question is constructed as “How many o_i are there?”, with the object class in its plural form for readability, and then converted into two generic question types as above.

Statistics. The **PointQA-LookTwice** dataset contains 57,405 questions across 34,676 images, including (1) *train* with 37,981 human-written and 14,713 automatically-generated questions across 32,925 images, (2) *val* with 997 human-written questions across 380 images, and (3) *test* with 3,714 human-written questions across 1,371 images.

The answer distribution is reasonably balanced: the answers “1”, “2”, and “ > 2 ” appear with 35.4%, 33.7%, and 30.9% frequency respectively in the test set. The most common objects in the dataset are people (29.7%), cars (3.1%), and signs (2.9%). Due to the high representation of people, the most common super-category is beings (48.9%), followed by objects (36.7%) and vehicles (14.3%). Given only the object name, a model can achieve an accuracy of at most 42.1%, indicating that visual reasoning is necessary. Example questions are shown in Fig. 5.

Human accuracy. As in Sec. 3.1 we run a small-scale human study on 100 *test* questions to evaluate the reliability of the data. Humans agreed with the answer 79% of the time, suggesting that the counting task with a point is doable. Please see Appendix A.3 for details on human evaluation.

4.2. PointQA-LookTwice model

Local-only. We adapt the Pythia-based model of Sec. 3.2 to this new setting, producing a model that combines local with global contextual reasoning. So far we had a set of visual features $\{\mathbf{v}_i^{pt}\}$ corresponding to region proposals containing the point pt , and we computed an attention vector over these proposals. Concretely, we projected the visual features $\{\mathbf{v}_i^{pt}\}$ and the text features of the question q to a common space, multiplied them element-wise and then ran through fully-connected layers to produce an attention vector, which is then normalized to a probability distribution \mathbf{a}^{pt} . This allowed us to produce $\mathbf{v}^{pt} = \sum_i \mathbf{a}_i^{pt} \mathbf{v}_i^{pt}$ describing the relevant information of the local region.

Global attention. Now we further use this local information to attend to the entire image: concretely, we project the visual features $\{\mathbf{v}_i\}$ from *all* region proposals, the text features q , and the new \mathbf{v}^{pt} to a common subspace, and then as before element-wise multiply and run through fully-connected layers to compute the normalized attention \mathbf{a}^{all} , yielding the global representation $\mathbf{v}^{all} = \sum_i \mathbf{a}_i^{all} \mathbf{v}_i$. The intuition is that \mathbf{v}^{pt} extracts local information relevant to the question, which is then used to attend to relevant regions in the whole image. Given these three streams of information, q , \mathbf{v}^{pt} and \mathbf{v}^{all} , we combine them through pairwise element-wise multiplication and concatenation (multiply q and \mathbf{v}^{pt} , and q and \mathbf{v}^{img} , and concatenate the two results) followed by fully connected layers and a softmax output (step (g) in Sec. 3.2).

4.3. PointQA-LookTwice evaluation

We benchmark the new global model trained on the *train* set of PointQA-LookTwice using the setup from Sec. 3.3.

Test accuracy. When trained to answer the question “How many of *these* are there?” along with a disambiguating point input, the model achieves an accuracy of 56.5%, significantly higher than an image-only version (without the point) at 46.1% and the modal answer (“1”) baseline at 35.4%. As expected it is somewhat behind having access to the object’s bounding box at 59.1%. In Table 2 we demonstrate how the accuracy changes with decreasing the ambiguity of the question: specifying the supercategory (being, vehicle or object) boosts the accuracy from 56.5% to 59.1%, and naming the object class further boosts it to 62.8% (in fact, as expected making the need for spatial supervision irrelevant, as the image-only model achieves 62.7% in this setting).

Attention analysis. We examine the model trained on supercategory questions. There are two types of attention: attention around the point is relatively peaked, with an average max attention of 0.42; as expected, global attention on the image is diffuse with an average max attention of 0.07. An example result is shown in Fig. 6.

	Spatial Disambiguation			Prior
	None	Point	Box	
How many of <i>these</i> ...	46.1	56.5	59.1	37.8
How many of <i>these</i> [supercategory]...	53.1	59.1	60.7	38.6
How many [object]...	62.7	62.8	62.5	40.3

Table 2: Results of the Pythia-based global model (Sec. 4.2) on the **PointQA-LookTwice test**. Rows are the question asked; columns are the disambiguation provided. Prior is a language-only model.



Figure 6: Attention of the Pythia-based model on a question in the **PointQA-LookTwice test** set. Point is in red.

Global vs local-only attention. We demonstrate the need for the new global attention added in Sec. 4.2 by comparing with the previous model which only considers the local regions around the point. The local-only model achieves an accuracy of 55.9% on supercategory questions, significantly lower than the global model at 59.1%. As expected, this is primarily due to under-counting (since the local-only model is not attending to other object instances). Concretely, on questions with the answer “1”, the global model performs slightly better than the local-only model (75.64% vs. 74.12%); however the difference is greater for answer ≥ 2 , where the global model achieves 50.04% and local model only 45.84%. Finally, as a sanity-check, we verified that the models achieve equal accuracy on the simpler **PointQA-General** dataset, indicating that the global model is well-suited for a wider range of pointing-based questions.

5. PointQA-General: generalized reasoning from a point input

Equipped with the insights of Sec. 3 and 4, where we examined well-structured pointing questions, we now turn our attention to the unconstrained setting where the questions become much more complex and the models must reason about image, textual, and spatial input in full generality. We construct a new dataset, modify state-of-the-art transformer-based models, and perform analysis in this new setting to round out our exploration.



Figure 7: Examples entries in the **PointQA-General** Dataset. Point is indicated in red.

5.1. PointQA-General dataset

We generate the **PointQA-General** dataset by adapting the human-written questions from Visual7W [31].

Question Generation. “Which” questions in Visual7W are paired with bounding boxes as multiple choice answers; for example, they might ask “Which pillow is closest to the window?”, where one of the four provided boxes is correct. We transform these questions into pointing-QA using the following formula (where X is the subject and Y is a description):

- “Which X is Y ” becomes “Is this X Y ? ”
- “Which X are Y ” becomes “Are these X Y ? ”
- “Which X has Y ” becomes “Does this X have Y ? ”
- “Which X have Y ” becomes “Do these X have Y ? ”

A high proportion of human-written questions follow one of these templates; questions that do not are not included. From each included question in Visual7W we generate two pointing questions: (1) using the correct bounding box, with the answer “yes” and (2) using one of the three incorrect boxes selected randomly with the answer “no”. The point input is simulated as the center of the bounding box. Example questions are shown in Fig. 7.

Statistics. The final dataset consists of 319,300 questions over 25,420 images, with the “yes” and “no” answers equally balanced. The dataset is divided into (1) *train* (255,072 questions across 20,337 images), (2) *val* (32,276 questions across 2,541 images), and (3) *test* (31,952 questions across 2,542 images). The most common subjects asked about are “part” (e.g. “Is this part of the desktop computer touching the carpet?”; 6.2% of the questions), “object” (4.4%), “person” (4.3%), “item” (2.7%), “man” (1.6%), “tree” (1.3%), “animal” (1.2%), and “food” (1.0%). The reference descriptions are quite complex: they contain 6 words on average (95% fall between 3 and 12 words) compared to the much simpler attributes of **PointQA-Local** which are typically only 1 word long.

Human accuracy. Human accuracy on 100 random *test* questions is 91%; this confirms both our assumptions about simulating the point as the center of the bounding box as well as the reliability of the overall data collection pipeline.

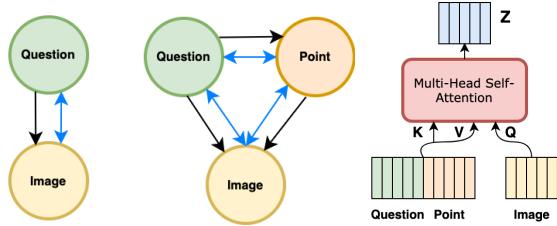


Figure 8: Left shows the different attention dependency structures between image and text; middle shows when the “point stream” is added (Sec. 5.2). Blue arrows indicate the bidirectional structure and black arrows unidirectional. Right shows the exact way in which cross-attention is implemented for transformer models with three streams.

Please see Appendix A.1 for details on human evaluation.

5.2. PointQA-General models

Given the increased complexity of the questions in **PointQA-General** we adapt two recent transformer-based VQA models to our pointing-based VQA task, MCAN [28] and LXMERT [27] (in addition to the Pythia-based model of Sec. 4.2). At a high-level MCAN computes a unidirectional attention like Pythia where the text influences attention over the image; by contrast in the bidirectional approach of LXMERT [27] each modality influences attention over the other (Fig. 8a). We now describe the model details.

MCAN-based. MCAN [28] is the winner of the the 2019 VQA Challenge. It works as follows: (1) The question q is broken into tokens $\{q_j\}$ and is encoded by stacking several self-attention layers $\{q_j\}_{\ell+1} = A(Q_\ell, K_\ell, V_\ell)$, where A is the self-attention mechanism and (Q_ℓ, K_ℓ, V_ℓ) are the queries, keys, and values derived from $\{q_j\}_\ell$ at layer ℓ . (2) The image representation over visual features derived from region proposals $\{\mathbf{v}_j\}$ is computed using a stack of self-attention and cross-attention layers. In each cross-attention layer, the keys and values are derived from the question representation at final layer L ; in other words, $\{\mathbf{v}_j\}_{\ell+1} = A(Q_\ell, W_K \{q_j\}_L, W_V \{q_j\}_L)$ for weights W_K and W_V corresponding to keys and values respectively. (3) The feature representations at the final layer $\{q_j\}_L$ and $\{\mathbf{v}_j\}_L$ are pooled, fused, and fed into a classifier.

To adapt this model to further accept the point input, we compute visual features $\{\mathbf{v}_j^{pt}\}_L$ corresponding only to the regions containing the point pt (the “point stream”), conditioned on the question as in the original model. Then in each cross-attention layer for the *image* stream, we condition attention on information from both the question and point by concatenating the representations $\{f_j\}_\ell = \{q_j\}_\ell \oplus \{\mathbf{v}_j^{pt}\}_\ell$ and then computing the keys and values (Fig. 8c). We then compute the multimodal fusion function:

$$\begin{aligned} z_1 &= \text{LayerNorm}(W_1^T \{q_j\}_L + W_2^T \{\mathbf{v}_j^{pt}\}_L), \\ z_2 &= \text{LayerNorm}(W_1^T \{q_j\}_L + W_3^T \{\mathbf{v}_j\}_L), \\ z &= z_1 \oplus z_2, \end{aligned}$$

followed by a linear classifier over the answer space. This dependency between streams is shown in Fig. 8b (black).

LXMERT-based. LXMERT implements a bidirectional approach to attention; first, the question and visual features are passed through a stack of self-attention layers to obtain $\{q_j\}_L$ and $\{\mathbf{v}_j\}_L$ as above. Then both are passed through a stack of “cross-modality encoders” that include a cross-attention layer followed by a self-attention layer and small feed-forward layer. The cross-attention layer exchanges the keys and values of the other modality (exactly as MCAN but for *both* image and text).

We modify LXMERT similarly to MCAN to accept the point input: it now has three streams corresponding to the question \mathbf{q} , the global image stream corresponding to all image regions \mathbf{v} , and a local point stream corresponding to only the regions \mathbf{v}^{pt} . Concretely, for the cross-attention layer at layer ℓ operating on the visual features $\{\mathbf{v}_j\}_\ell$, we concatenate $\{f_j\}_\ell = \{q_j\}_\ell \oplus \{\mathbf{v}_j^{pt}\}_\ell$ and compute keys and values on top of this representation; we do this similarly for the other modalities at layer ℓ . Thus, at each layer each modality influences attention over the other in the cross-attention modules (Fig. 8b, blue). As in LXMERT the final language features are pooled and fed into a classifier.

Two-Stream vs. Three-Stream. An alternative way of incorporating both local and contextual information is to concatenate the image features $\{\mathbf{v}_j\}$ and the point features $\{\mathbf{v}_j^{pt}\}$ into the same stream without modifying the VQA model itself. We explore this alternative in Sec 5.3 which we term **two-stream**, compared to the **three-stream** approach we describe in this section that treats the point as its own stream and modifies the cross-attention accordingly.

5.3. PointQA-General evaluation

We evaluate the Pythia-based [29] model of Sec. 4.2 and the MCAN-based [28] and LXMERT-based [27] models of Sec. 5.2 on the challenging task of **PointQA-General**.

Implementation Details.. For MCAN we set $L = 2$, since we found that increasing number of layers did not improve performance. For LXMERT following the original paper, we set a higher number of layers for the language modality; concretely we set $N_L = 5$ and $N_{Img} = N_{Pt} = 3$. We set the number of cross-modality encoders N_X to be 3. Both models were trained using Adam with early stopping and a patience of 5000 iterations. We used a learning rate of $\{5, 2.5\}e-5$ for MCAN and LXMERT respectively with a warmup/decay schedule. Details for Pythia are in Sec. 3.3.

	Pythia	MCAN	LXMERT
Q-Only	50.00	50.00	50.00
Image+Q	50.00	50.00	50.00
Point+Q	81.81	82.60	81.33
Two-Stream	77.84	81.62	82.41
Three-Stream	83.12	83.21	81.71

Table 3: Accuracy on the PointQA-General dataset. We evaluate the following methods and baselines: (1) Three-Stream and Two-Stream as described in Sec 5.2, (2) *Q-only* relies only on the language of the question; (3) *Image+Q* provides the question and visual features $\{\mathbf{v}_j\}$ corresponding to the entire image, and (4) *Point+Q* provides the question and only the features $\{\mathbf{v}_j^{pt}\}$ for the region proposals containing the point (for Pythia this is the local-only model of Sec. 3). Note that all these baselines are outperformed by our proposed models which capture local and global contextual information, and that the highest accuracy is achieved with our three-stream approach on the MCAN model.

Test Results. Results on the **PointQA-General** dataset are shown in Table 3. The single highest accuracy is achieved using our three-stream approach with the MCAN-based model. Across all models, the three-stream approach outperforms any of the ablations that use one (*Q-Only*) or two streams (*Image+Q*, *Point+Q*). The high overall accuracy of our three-stream approach and its improvement over *Point+Q* indicates the benefit of adding the point as a separate stream and modifying the model cross-attention to create a rich set of contextual interactions between the streams. The *Image+Q* and *Q-Only* models perform no better than random chance since for each image-question pair in Visual7W we generate two questions with opposite answers.

The MCAN-based method appears to achieve the strongest performance, even for the *Point+Q* ablation; this suggests it might make more effective use of the limited contextual information available in this setting. For both MCAN and Pythia, the two-stream approach makes less effective use of contextual information than three-stream. Surprisingly, the LXMERT model achieves higher accuracy on the two-stream approach. One possible reason is that the pooling strategy used by LXMERT only uses the language representation; it is possible that pooling from all three streams would improve performance of the three-stream method. Qualitative results are shown in Fig. 9.

Using Image Context. Certain questions in the **PointQA-General** Dataset require reasoning about the image context beyond a local region around the point, such as questions involving comparison (e.g. “Is this zebra the *most* obstructed from view?”) or directional reasoning (“Are these crayons on a table *near* a cat?”). Viewing only the object being directly pointed to is insufficient to answer these ques-



Figure 9: Examples in the **PointQA-General** test-set where our three-stream approach (w/ Pythia) is correct and the Point+Q ablation does not have contextual information.

tions. The accuracy improvement of the three-stream model over the *Point+Q* ablation is particularly significant for such questions containing comparison/directional words; e.g. for questions with the word ‘farthest’, the accuracy difference of the two models (w/ Pythia) is 3.66% vs. 1.81% overall. This result holds strongly across several such words, suggesting that the improved performance of the three-stream model is due to its ability to effectively incorporate the broader image context.

6. Conclusion

In summary, we introduced three novel types of visual questions *requiring* a spatial point input for disambiguation. For each question type we created a benchmark dataset and model design and performed extensive analysis. We hope our work inspires further research in this space of questions.

7. Acknowledgements

This work is partially supported by Samsung and by the Princeton SEAS Project-X Funding Award. Thank you to Karthik Narasimhan, Zeyu Wang, Felix Yu, Zhiwei Deng, and Deniz Oktay for helpful discussions and feedback on this work. We would also like to thank Sunnie Kim, Zeyu Wang, Sharon Zhang, Angelina Wang, Nicole Meister, Dora Zhao, Ozge Yalcinkaya, Vikram Ramaswamy, and Anat Kleiman for participating in the human evaluations.

References

- [1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don’t Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition*, 2018. [2](#)
- [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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [3](#)
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. [1](#)
- [4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What’s the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*, 2016. [2](#)
- [5] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. 2015. [2](#)
- [6] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [1, 2](#)
- [7] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [1](#)
- [8] Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. VQS: Linking Segmentations to Questions and Answers for Supervised Attention in VQA and Question-Focused Semantic Segmentation. In *International Conference on Computer Vision (ICCV)*, 2017. [2](#)
- [9] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [2](#)
- [10] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. 2019. [1, 2](#)
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [4](#)
- [12] Michael Hild, Motonobu Hashimoto, and Kazunobu Yoshida. Object recognition via recognition of finger pointing actions. In *Image Analysis and Processing*, 2003. [2](#)
- [13] Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [1, 2](#)
- [14] Suyog Dutt Jain and Kristen Grauman. Click carving: Segmenting objects in video with point clicks. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2016. [2](#)
- [15] Huaiyu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#)
- [16] 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. In *International Journal of Computer Vision*, 2017. [1, 2, 3, 4, 5](#)
- [17] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. 2017 *IEEE Conference*

- on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 4
- [18] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2
- [19] Bertram F Malle, Louis J Moses, and Dare A Baldwin. *Intentions and intentionality: Foundations of social cognition*. MIT press, 2001. 1, 2
- [20] David Merrill and Pattie Maes. Augmenting looking, pointing and reaching gestures to enhance the searching and browsing of physical objects. In *Pervasive Computing*. 2007. 2
- [21] John Oates and Andrew Grayson. *Cognitive and language development in children*. Blackwell; Open University Press, 2004. 1, 2
- [22] Dim P. Papadopoulos, Alasdair D. F. Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *European Conference of Computer Vision (ECCV)*, 2014. 2
- [23] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [24] Syed Shaukat Raza Abidi, MaryAnn Williams, and Benjamin Johnston. Human pointing as a robot directive. In *International Conference on Human-robot interaction*, 2013. 2
- [25] Shaqiq Ren, Kaiming He, Ross B. Girshick, and J. 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. 3
- [26] Allison Sauppe and Bilge Mutlu. Robot deictics: how gesture and context shape referential communication. In *International Conference on Human-Robot Interaction*, 2014. 2
- [27] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. 2, 7, 8
- [28] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 7, 8
- [29] Yu Jiang*, Vivek Natarajan*, Xinlei Chen*, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. 2018. 2, 3, 4, 5, 8
- [30] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable Visual Question Answering by Visual Grounding From Attention Supervision Mining. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2
- [31] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded Question Answering in Images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 7

A. Human Evaluations

We conduct human evaluations of the **PointQA-Local**, **PointQA-LookTwice**, and **PointQA-General** datasets to check quality and ensure the reliability of our data collection pipeline. Our methodology is as follows. We select 100 questions at random from each of the test subsets of the three datasets. For each dataset, we ask three volunteers to answer the 100 questions sampled from that dataset; thus our study consists of a total of 9 participants distributed evenly across the three datasets. We send each participant a Jupyter notebook displaying the set of 100 questions and provide the following instructions for each question:

Please view the image and the small red point in the image, and answer the question to the best of your ability. The red point indicates what the question is asking about. Type in the number corresponding to the answer. Click [Enter] to move on to the next question.

We received 300 responses per dataset (three annotators each for the 100 questions). We record the overall human accuracy (with the denominator being the 300 responses) and for each incorrect response, the potential reason to understand common failure modes in our datasets.

A.1. PointQA-General

We begin with the main benchmark we introduce in this paper, the **PointQA-General** Dataset of 319,300 questions across 25,420 images. Human accuracy across the 100-question subset (300 annotator answers) is **90.7%**. This high accuracy indicates that our question construction and point generation process yields sensible point-input questions that can be reasonably answered by humans. The accuracy of the three-stream model w/ Pythia (Sec. 4.2) on this 100-question subset is 82.0%; the substantial human-machine performance gap indicates that the **PointQA-General** dataset is a challenging benchmark for point-input VQA and there is still scope for further model development in this question space.

For the questions in this subset where human annotators disagreed with our dataset answers, we analyze and categorize the reasons for disagreement. The most common are:

1. “Other reasonable answer” (4.0%). Another answer in the dataset - in this case *the other* answer since yes/no questions - was a reasonable choice for the question. These correspond to questions in Visual7W that are challenging for human annotators.
2. “Point not on object” (2.3%). The location of the point is incorrect and the point does not refer to the object

being asked about. The fact that this accounts for a very small fraction of errors confirms that drawing the point at the center of the bounding box is a reasonable approach overall.

- “Wrong attention” (1.3%). The human annotator pays attention to the wrong object when answering the question. This sometimes occurs when a bounding box answer that is incorrect in Visual7w refers to another object than is asked about in the question, so the annotator pays attention to the correct bounding box. However again this accounts for a small proportion of errors in the dataset.

The remaining disagreements are “question doesn’t make sense” (1.0%), no obvious reason (0.3%) and obstruction (0.3%).

A.2. PointQA-Local

Human accuracy on the 100 question subset of the **PointQA-Local** Dataset is 75.7%; this indicates that our questions can be reasonably answered by human annotators. By comparison the accuracy of the **PointQA-Local** model (Section 3.2) on this question subset is actually 82.0%; this can be reasoned by the fact that annotators are presented with a larger set of answers (20) in **PointQA-Local** and when annotating quickly can often choose answers that are reasonable but ‘less correct’ (as discussed below). By contrast in the more robust **PointQA-General** setting the annotators have fewer answer choices and the human-machine gap is a result of the complexity of the question language and higher-level reasoning involved.

We further analyze the most common form of errors on the **PointQA-Local** Dataset. Similar to the **PointQA-General** Dataset, the human errors come from “other reasonable answer” (12.3%), “ambiguous point” (4.3%), “obstruction” (3.3%), “wrong attention” (2.3%), “answer misannotation (1.0%) and “no obvious answer” (1.0%). “Ambiguous point” refers to cases where it is ambiguous which object is being referred to by the point. “Obstruction” refers to cases where the object in question is obstructed by another object in the image or the point. Qualitative examples are shown in Fig. 10. The prevalence of the “other reasonable answer” category indicates that Visual Genome annotations from which we derived our questions can be accurate but incomplete (for example in Fig. 10 on the left, the coat is most accurately labeled gray and not brown, although both labels could be reasonably included in the annotations).

A.3. PointQA-LookTwice

Human accuracy on the 100-question subset of the **PointQA-LookTwice** Dataset is 79.3%; by comparison the global model of Sec. 4.2 achieves 77.0%. The most common failure modes are “answer misannotation” (7.7%; i.e., the answer in the dataset is incorrect), “variable local-to-



Figure 10: Examples of common failure modes in the **PointQA-Local** Dataset.



Figure 11: Examples of common failure modes in the **PointQA-LookTwice** Dataset.

global reasoning” (6.7%; i.e., the human annotator considered the object pointed to as more generic or specific than intended in the question, and thus over/under-counted), “other instances hard to see” (3.3%), “wrong attention” (1.3%), “ambiguous point” (1.0%), and “no obvious reason” (0.7%). “Variable local-to-global reasoning” is an important failure mode since replacing the object name with a supercategory (e.g. “these objects”) might cause the annotator to count incorrectly; however, the high overall accuracy indicates this is a not a widespread issue. Qualitative examples are shown in Fig. 11. In the study, we provided the annotators with possible answer choices of 1 through 7; when evaluating the responses, we narrowed the choices down to 1, 2, >2.

Finally, we note that questions were unintentionally drawn from the training set of the **PointQA-LookTwice** Dataset. This does not fundamentally affect the quality of our human studies. The model accuracy number of 77.0% is when excluding these questions from the training set, but could still be inflated. One thing to note is that the source of human disagreement resulting from misannotation in the dataset would be substantially lower without questions from the training set (since the test set consists exclusively of human-written questions).