From Recognition to Cognition: Visual Commonsense Reasoning

Rowan Zellers[♠] Yonatan Bisk[♠] Ali Farhadi[♠♡] Yejin Choi[♠♡]
[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington
[♡]Allen Institute for Artificial Intelligence
visualcommonsense.com

Figure 1: VCR: Given an image, a list of regions, and a question, a model must answer the question and provide a *rationale* explaining why its answer is right. Our questions challenge computer vision systems to go beyond recognition-level understanding, towards higher-order cognitive and commonsense understanding of the world depicted by the image.

Abstract

*Visual understanding goes well beyond object recognition. With one glance at an image, we can effortlessly imagine the world beyond the pixels: for instance, we can infer people's actions, goals, and mental states. While this task is easy for humans, it is tremendously difficult for today's vision systems, requiring higher-order cognition and commonsense reasoning about the world. In this paper, we formalize this task as **Visual Commonsense Reasoning**. In addition to answering challenging visual questions expressed in natural language, a model must provide a rationale explaining why its answer is true. We introduce a new dataset, VCR, consisting of 290k multiple choice QA problems derived from 110k movie scenes. The key recipe to generating non-trivial and high-quality problems at scale is **Adversarial Matching**, a new approach to transform rich annotations into multiple choice questions with minimal bias. To move towards cognition-level image understanding, we present a new reasoning engine, called Recognition to Cognition Networks (**R2C**), that models the necessary layered inferences for grounding, contextualization, and reasoning. Experimental results show that while humans find VCR easy (over 90% accuracy), state-of-the-art models struggle (∼45%). Our **R2C** helps narrow this gap (∼65%); still, the challenge is far from solved, and we provide analysis that suggests avenues for future work.*

1. Introduction

With one glance at an image, we can immediately infer what is happening in the scene beyond what is visually obvious. For example, in the top image of Figure 1, not only do we see several objects (people, plates, and cups), we can

also reason about the entire situation: three people are dining together, they have already ordered their food before the photo has been taken, [person3] is serving and not eating with them, and what [person1] ordered are the pancakes and bacon (as opposed to the cheesecake), because [person4] is pointing to [person1] while looking at the server, [person3].

Visual understanding requires seamless integration between *recognition* and *cognition*: beyond recognition-level perception (e.g., detecting objects and their attributes), one must perform cognition-level reasoning (e.g., inferring the likely intents, goals, and social dynamics of people) [15]. State-of-the-art vision systems can reliably perform *recognition*-level image understanding, but struggle with complex inferences, like those in Figure 1. We argue that as the field has made significant progress on recognition-level building blocks, such as object detection, pose estimation, and segmentation, now is the right time to tackle cognition-level visual commonsense reasoning at scale.

As a critical step toward complete visual understanding, we present a new task formulation, **Visual Commonsense Reasoning**. Given an image, a model must answer a question that is and requires a thorough understanding of the global context of the visual world evoked by the image. Moreover, a model must provide a rationale justifying why that answer is true, referring to the details of the scene, as well as background knowledge about how the world works. These questions, answers, and rationales are expressed using a mixture of rich natural language as well as *references* to image regions detected by state-of-the-art object detection systems [25, 30]. To support clean-cut evaluation, all our tasks are framed as multiple choice QA.

Our new dataset for this task, VCR, is the first of its kind and is large-scale — 290k pairs of questions, answers, and rationales, over 110k unique movie scenes. A crucial challenge in constructing a dataset of this complexity at this scale is how to avoid annotation artifacts; a recurring challenge in most recent QA datasets has been that human-written answers contain unexpected but distinct biases that models can easily exploit. Often these biases are so prominent so that models can select the right answers without even looking at the questions [29, 62, 73].

Thus, we present **Adversarial Matching**, a novel QA assignment algorithm that allows for robust multiple-choice dataset creation at scale. The key idea is to recycle each correct answer for a question exactly three times — as a negative answer for three other questions. Because each answer appears exactly four times in the dataset with the equal probability 25% of being correct, the model cannot get lucky based on the annotation artifacts in answers. We formulate the answer recycling problem as a constrained optimization based on the relevance and entailment scores between each candidate negative answer and the gold answer, as measured by state-of-the-art natural language inference models [11, 58, 17]. A neat feature of our recycling algorithm is a knob that can control the tradeoff between human and machine difficulty: we want the problems to be hard for machines while easy for humans.

Narrowing the gap between recognition- and cognition-level image understanding requires grounding the meaning of the natural language passage in the visual data, understanding the answer in the context of the question, and reasoning over the shared and grounded understanding of the question, the answer, the rationale and the image. In this paper we introduce a new model, **Recognition to Cognition Networks** (**R2C**). Our model performs three inference steps. First, it *grounds* the meaning of a natural language passage with respect to the image regions (objects) that are directly referred to. It then *contextualizes* the meaning of an answer with respect to the question that was asked, as well as the global objects not mentioned. Finally, it *reasons* over this shared representation to arrive at an answer. Our experiments show that our model outperforms state-of-the-art visual question-answering systems by a large margin, obtaining 65% accuracy at question answering, 67% at answer justification, and 44% at joint answering and justification on VCR. Still, the task and dataset is far from solved: humans score about 90% on each. We provide detailed insights and an ablation study to point to avenues for future research.

In sum, our major contributions are fourfold: (1) we formalize a new task, Visual Commonsense Reasoning, and (2) present a large-scale multiple-choice QA dataset, VCR, (3) that is automatically assigned using **Adversarial Matching**, a new algorithm for robust multiple-choice dataset creation. (4) We also propose a new model, **R2C**, that aims to mimic the layered inferences from recognition to cognition, and establish baseline performance on our new challenge. The dataset is available to download, along with code for our model, at visualcommonsense.com.

2. Task Overview

We present VCR, a new task that challenges vision systems to holistically and cognitively understand the content of an image. For instance, in Figure 1, we need to understand the activities ([person3] is delivering food), the roles of people ([person1] is a customer who previously ordered food), the mental states of people ([person1] wants to eat), and the likely events before and after the scene ([person3] will deliver the food and leave the table). Our task covers these categories and more: a distribution of the inferences required is in Figure 2.

Fundamental to achieving visual understanding, however, is for a classifier to be able to choose the right answer *for the right reasons*. We accomplish this by requiring a model to give a natural-language *rationale* that explains why its answer is true. Our questions, answers, and ratio-

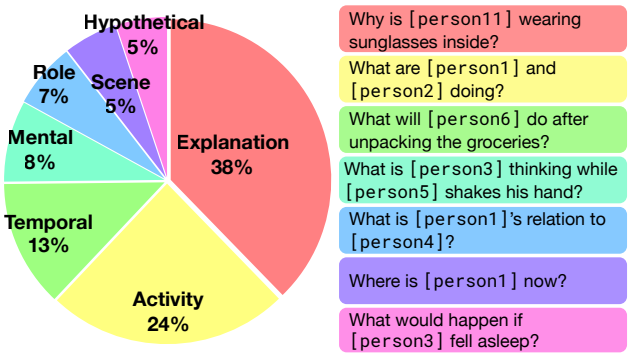


Figure 2: Overview of the types of inference required by questions in VCR. Of note, 38% of the questions are explanatory 'why' or 'how' questions, 24% involve recognizing cognition-level activities, and 13% require reasoning over time (i.e., inferring what might come next). Note that these categories are not mutually exclusive: much of the dataset falls into several categories, often requiring multiple hops of inference to arrive at the right answer. More details regarding the categories are in the appendix, Sec A.

nales are written in a mixture of rich natural language as well as detection tags, like '[person2]': this helps to provide an unambiguous link between the textual description of an object ('the man on the left in the white shirt') and the corresponding image region. Evaluation is straightforward, as our dataset is entirely multiple choice.

For our task, we consider the following three modes:

- $Q \rightarrow A$: given a question, select the correct answer.
- $QA \rightarrow R$: given a question and the correct answer, select the correct rationale.
- $Q \rightarrow AR$: given a question, select the correct answer, then the correct rationale. The model prediction is correct only if both the answer *and the rationale* are correct, helping alleviate concerns that a model answers correctly, but for questionable reasons.

Formal Task Definition More formally, our task is defined as follows. A model is given an image $\boldsymbol{I}$, and:

- A sequence $\boldsymbol{o}$ of object detections. Each object detection o_i consists of a *bounding box* $\mathbf{b}$, a segmentation mask $\mathbf{m}$[1], and a class label $\ell_i \in \mathcal{L}$.
- A *query* $\boldsymbol{q}$, posed using a mix of natural language and pointing. Each word $\boldsymbol{q}_i$ in the query is either a word in a vocabulary $\mathcal{V}$, or is a tag referring to an object in $\boldsymbol{o}$.
- A set of *responses*, where each response $\boldsymbol{r}^{(i)}$ is written in the same manner as the query - a mixture of natural language and pointing. In our paper, we use N=4 responses, of which exactly one is correct.

[1]The task is agnostic to the representation of the mask, but it could be thought of as a list of polygons $\boldsymbol{p}$, with each polygon consisting of a sequence of 2d vertices inside the box $\boldsymbol{p}_j = \{x_t, y_t\}_t$.

The model's goal is to predict which of the N=4 responses is the correct one, and we thus evaluate models in terms of accuracy, where the baseline performance is given by $1/N$. For $Q \rightarrow AR$, the baseline performance is $1/N^2$, as we perform this procedure twice: once for the answer, then once for the rationale.

3. Related Work

Question Answering Visual Question Answering [5] was one of the first large-scale datasets that framed visual understanding as a QA task, with questions about COCO images [50] typically answered with a short phrase. This line of work also includes 'pointing' questions [46, 94] and templated questions with open ended answers [87]. Recent datasets also focus on knowledge-base style content [81, 84]. On the other hand, the answers in VCR are entire sentences, and the knowledge required by our dataset is largely background knowledge about how the world works.

Recent work also includes movie or TV-clip based QA [76, 52, 47]. In these settings, a model is given a video clip, often alongside additional language context in the form of subtitles, a movie script, or the plot summary. This language context makes the task much easier - VCR, in contrast, features no extra language context besides the question. Additionally, our dataset is simple to approach: there is no need to perform person identification [67] or linkage with subtitles.

An orthogonal line of work has been on referring expressions: asking to what image region a natural language sentence refers to [61, 53, 66, 88, 89, 60, 37, 34]. We explicitly avoid referring expression-style questions by using indexed detection tags (like '[person1]').

Last, some work focuses on commonsense phenomena, such as 'what if' and 'why' questions [80, 59]. However, the space of commonsense inferences is often limited by the underlying dataset chosen (synthetic [80] or COCO [59] scenes). In our work, we ask commonsense questions in the context of rich images from movies.

Explainability AI models are often right, but for questionable or vague reasons [7]. This has motivated work in having models provide explanations for their behavior, in the form of a natural language sentence [32, 10, 42] or an attention map [33, 36, 38]. Our rationales combine the best of both of these approaches, as they involve both natural language text as well as references to image regions. Additionally, while it is hard to evaluate the quality of generated model explanations, choosing the right rationale in VCR is a multiple choice task, making evaluation straightforward.

Commonsense Reasoning Our task unifies work involving reasoning about commonsense phenomena, such as physics [55, 85], social interactions [2, 78, 13, 28], procedure understanding [92, 3] and predicting what might happen next in a video [75, 19, 93, 79, 20, 65, 86].

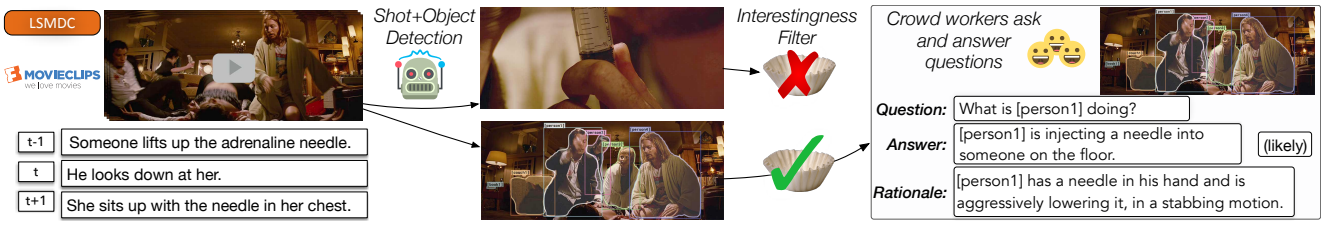


Figure 3: An overview of the construction of VCR. Using a state-of-the-art object detector [30, 25], we identify the objects in each image. The most interesting images are passed to crowd workers, along with scene-level context in the form of scene descriptions (MovieClips) and video captions (LSMDC, [68]). The crowd workers use a combination of natural language and object tags to ask and answer challenging visual questions, also providing a rationale justifying their answer.

Adversarial Datasets Past work has proposed the idea of creating adversarial datasets, whether by balancing the dataset with respect to priors [26, 29, 63] or switching them at test time [1]. Most relevant to our work is SWAG [90], a multiple choice NLP dataset where the correct answers are human-written, while wrong answers are chosen from a pool of machine-generated text that is further validated by humans. However, the correct and wrong answers are from fundamentally different sources, raising the concern that models are performing authorship identification rather than deep reasoning. In contrast, in **Adversarial Matching**, the wrong choices come from the same distribution as the right choices, and no human validation is required.

4. Data Collection

In this section, we describe how we construct VCR. Our goal is to collect commonsense visual reasoning problems at scale, while ensuring diverse and high-quality inferences.

Interesting and Diverse Situations To ensure diversity, we make no limiting assumptions about the predefined set of actions or objects [77, 18, 22]: rather than searching for predefined labels, which can introduce search engine bias, we collect data from a diverse set of publicly available videos: the Large Scale Movie Description Challenge [68] and YouTube movie clips.[2] We trained and applied an interesting filter to avoid simple scenes (e.g. a closeup of a person holding a syringe in Figure 3).[3]

We center our task around challenging questions involving cognitive-level reasoning, explicitly avoiding referring expressions. For example, instead of annotating a low-level action: 'the guy who is raising his hand.' we instead wish to capture the social commonsense: 'he wants to ask a question'. To accomplish this, our interface provides annotators with object detections from Mask-RCNN [30, 25].[4] We allow workers to refer to detection tags (like [person3 ])

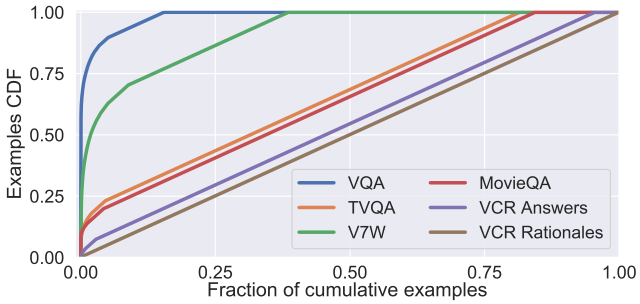


Figure 4: CDF of dataset examples ordered by frequency in question-answering datasets [5, 94, 76, 47]. To obtain this plot, we sampled 10,000 answers from each dataset (or rationales, for 'VCR rationales'). We consider two examples to be the same if they exactly match, after tokenization, lemmatization, and removal of stopwords. Where many datasets in this space are light-tailed, our dataset shows great diversity (e.g. almost every rationale is unique.)

when annotating the data. Workers are also provided with context in the form of video captions, helping them ask about and answer what will happen next.

Crowdsourcing Quality Annotations We used Amazon Mechanical Turk for crowdsourcing. Workers were given an image with detections and asked to provide one to three questions about the image. For each question, they supplied an answer as well as a rationale.[5] To avoid annotation artifacts and ensure top-tier work, we used a system of quality checks and paid our workers well.[6]

The result is an underlying dataset with high agreement and diversity of reasoning. Our dataset contains a myriad of interesting commonsense phenomena (Figure 2) and a great diversity in terms of unique examples (Figure 4): almost every answer and rationale is unique.

[2] We downloaded 30k videos from the Fandango MovieClips channel, available at youtube.com/user/movieclips.

[3] We annotated images for 'interestingness' and trained a classifier using CNN features and detection statistics, details in the appendix, Sec B.

[4] All images have at least three detections that workers can reference.

[5] Workers also selected, for each question, whether their answer is likely (>75% probability), possible (25-75%), or unlikely (<25%).

[6] More details in the appendix, Sec B.

5. Adversarial Matching

We cast VCR as a four-way multiple choice task, to avoid the evaluation difficulties of language generation or captioning tasks where current metrics often incorrectly score machines better than humans [50].[7] However, it is not obvious how to obtain high-quality incorrect choices, or counterfactuals, at scale. While past work has asked humans to write several counterfactual choices for each correct answer [76, 47], this process is expensive. Moreover, it has the potential of introducing annotation artifacts: subtle patterns that are by themselves highly predictive of the 'correct' or 'incorrect' label [73, 29, 62].

In this work, we propose **Adversarial Matching**: a new method that allows for any 'language generation' dataset to be turned into a multiple choice test, while requiring minimal human involvement. Our key insight is that the problem of obtaining good counterfactuals can be broken up into two subtasks: the counterfactuals must be as **relevant** as possible to the context (so that they appeal to machines), while they cannot be overly **similar** to the correct response (so that they don't become correct answers incidentally). We balance between these two objectives to create a dataset that is challenging for machines, yet easy for humans.

Formally, our procedure requires two models: one to compute the relevance between a query and a response, P_{rel}, and another to computes the similarity between two answer choices, P_{sim}. Here, we employ state-of-the-art models for Natural Language Inference (BERT [17] and ESIM+ELMo [11, 58], respectively).[8] Then, given dataset examples $(\boldsymbol{q}_i, \boldsymbol{r}_i)_{1\leq i\leq N}$, we obtain a counterfactual for each $\boldsymbol{q}_i$ by performing maximum-weight bipartite matching [56, 41] on a weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, given by

$$\mathbf{W}_{i,j} = \log(P_{rel}(\boldsymbol{q}_i, \boldsymbol{r}_j)) + \lambda \log(1 - P_{sim}(\boldsymbol{r}_i, \boldsymbol{r}_j)). \quad (1)$$

Here, $\lambda>0$ controls the tradeoff between similarity and relevance.[9] To obtain multiple counterfactuals, we perform several bipartite matchings. To ensure that the negatives are diverse, however, during each iteration we replace the similarity term with the maximum similarity between a candidate response $\boldsymbol{r}_j$ and all responses currently assigned to $\boldsymbol{q}_i$.

Ensuring dataset integrity To guarantee that there is no question/answer overlap between the training and test sets, we split our full dataset (by movie) into 11 folds, each of which is individually run through Eq 1. Two were pulled aside for validation and testing.

[7] Constructing new metrics is an active line of research [9, 14].

[8] We finetune P_{rel} (BERT), on the annotated data (taking steps to avoid data leakage), whereas P_{sim} (ESIM+ELMo) is trained on entailment and paraphrase data - details in appendix Sec C.

[9] We tuned this hyperparameter by asking crowd workers to answer multiple-choice questions at several thresholds, and chose the value for which human performance is above 90% - details in appendix Sec C.

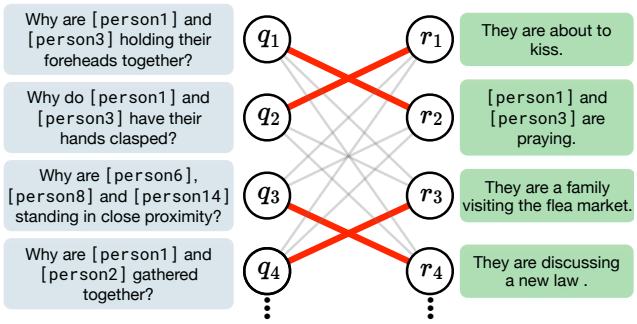


Figure 5: Overview of **Adversarial Matching**. Incorrect choices are obtained via maximum-weight bipartite matching between queries and responses; the weights are scores from state-of-the-art natural language inference models. Assigned responses are highly relevant to the query, while they differ in meaning versus the correct responses.

6. Recognition to Cognition Networks

We introduce Recognition to Cognition Networks (**R2C**), a new model for visual commonsense reasoning. To perform well on this task requires a deep understanding of language, vision, and the world. For example, in Fig 6, answering '*Why is* [person4] *pointing at* [person1]*?*' requires multiple inference steps. First, we **ground** the meaning of the query and each response, which involves referring to the image for the two people. Second, we **contextualize** the meaning of the query, response, and image together. This step includes resolving the referent 'he,' and why one might be pointing in a diner. Third, we **reason** about the interplay of relevant image regions: in this example, the model must determine the social dynamics between [person1] and [person4]. We formulate our model as three high-level stages: grounding, contextualizing, and reasoning, and use standard neural building blocks to implement each component.

In more detail, recall that a model is given an image, a set of objects $\boldsymbol{o}$, a query $\boldsymbol{q}$, and a set of responses $\boldsymbol{r}^{(i)}$ (of which exactly one is correct). The query $\boldsymbol{q}$ and response choices $\boldsymbol{r}^{(i)}$ are all expressed in terms of a mixture of natural language and pointing to image regions: notation-wise, we will represent the object tagged by a word w as o_w. If w isn't a detection tag, o_w refers to the entire image boundary. Our model will then consider each response $\boldsymbol{r}$ separately, using the following three components:

Grounding The grounding module will learn a joint image-language representation for each token in a sequence. Because both the query and the response contain a mixture of tags and natural language words, and we apply the same grounding module for each (allowing it to share parameters). At the core of our grounding module is a bidirectional LSTM [35] which at each timestep is passed as input a word representation for w_t, as well as visual fea-

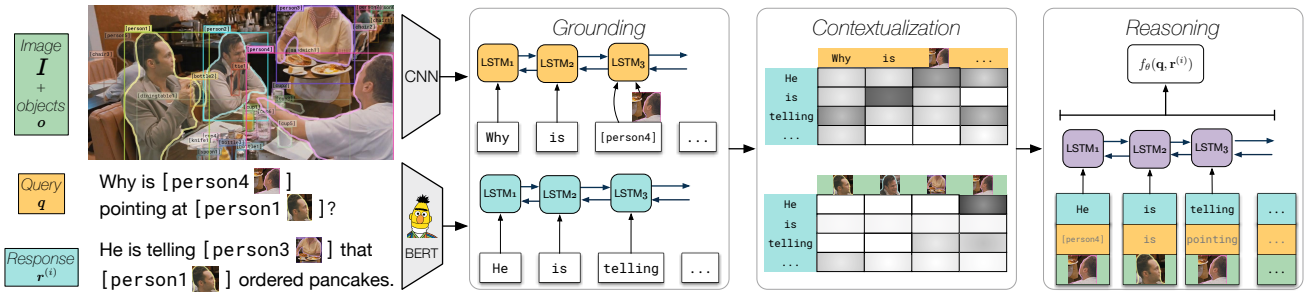


Figure 6: High-level overview of our model, **R2C**. We break the challenge of Visual Commonsense Reasoning into three components: grounding the query and response, contextualizing the response within the context of the query and the entire image, and performing additional reasoning steps on top of this rich representation.

tures for o_{w_t}. We use a CNN to learn object-level features: the visual representation for each region o_i is Roi-Aligned from its bounding region [64, 30]. To additionally encode information about the object's class label ℓ_i, we project an embedding of ℓ_i along with the object's visual features, into a shared hidden representation. Let the output of the LSTM at each position be $\mathbf{r}_i$, for the response and $\mathbf{q}_j$ for the query.

Contextualization Given a grounded representation of the query and response, we use attention mechanisms to contextualize these sentences with respect to each other and the image context. For each position i in the response, we will define the attended query representation as $\hat{\mathbf{q}}_i$ using the following equation:

$$\alpha_{i,j} = \underset{j}{\text{softmax}}(\mathbf{r}_i \mathbf{W} \mathbf{q}_i) \qquad \hat{\mathbf{q}}_i = \sum_j \alpha_{i,j} \mathbf{q}_j. \quad (2)$$

To contextualize an answer with the image, including implicitly relevant objects that have not been picked up from the grounding stage, we perform another bilinear attention between the response $\mathbf{r}$ and each object o_k's image features. Let the result of the object attention be $\hat{\mathbf{o}}_i$.

Reasoning Last, we allow the model to *reason* over the response, attended query and objects. We accomplish this using a bidirectional LSTM that is given as context $\hat{\mathbf{q}}_i$, $\mathbf{r}_i$, and $\hat{\mathbf{o}}_i$ for each position i. For better gradient flow through the network, we concatenate the output of the reasoning LSTM along with the question and answer representations for each timestep: the resulting sequence is max-pooled and passed through a multilayer perceptron.

Neural architecture and training details For our image features, we use ResNet50 [31]. To obtain strong representations for language, we used BERT representations [17]. BERT is applied over the entire question and answer choice, and we extract a feature vector from the second-to-last layer for each word. We train **R2C** by minimizing the multi-class cross entropy between the prediction for each response $\boldsymbol{r}^{(i)}$, and the gold label. See the appendix (Sec E) for detailed training information and hyperparameters.[10]

[10]Our code is also available online at visualcommonsense.com.

	Model	$Q \rightarrow A$ Val	$Q \rightarrow A$ Test	$QA \rightarrow R$ Val	$QA \rightarrow R$ Test	$Q \rightarrow AR$ Val	$Q \rightarrow AR$ Test
	Chance	25.0	25.0	25.0	25.0	6.2	6.2
Text Only	BERT	53.8	53.9	64.1	64.5	34.8	35.0
Text Only	BERT (response only)	27.6	27.7	26.3	26.2	7.6	7.3
Text Only	ESIM+ELMo	45.8	45.9	55.0	55.1	25.3	25.6
Text Only	LSTM+ELMo	28.1	28.3	28.7	28.5	8.3	8.4
VQA	RevisitedVQA [39]	39.4	40.5	34.0	33.7	13.5	13.8
VQA	BottomUpTopDown[4]	42.8	44.1	25.1	25.1	10.7	11.0
VQA	MLB [43]	45.5	46.2	36.1	36.8	17.0	17.2
VQA	MUTAN [6]	44.4	45.5	32.0	32.2	14.6	14.6
	R2C	**63.8**	**65.1**	**67.2**	**67.3**	**43.1**	**44.0**
	Human		91.0		93.0		85.0

Table 1: Experimental results on VCR. VQA models struggle on both question-answering ($Q \rightarrow A$) as well as answer justification ($Q \rightarrow AR$), possibly due to the complex language and diversity of examples in the dataset. While language-only models perform well, our model **R2C** obtains a significant performance boost. Still, all models underperform human accuracy at this task.

7. Results

In this section, we evaluate the performance of various models on VCR. Our goal is for models to not only make accurate commonsense inferences about the world, but to be able to justify their inferences in natural language. To achieve this requires a model to perform well on both modes in VCR: question answering ($Q \rightarrow A$) as well as answer justification ($QA \rightarrow R$). Thus, we also evaluate models' joint accuracy $Q \rightarrow AR$. In this mode, a model must choose the right answer for a question (given four answer choices), and then choose the right rationale for that question and answer (given four rationale choices). If it gets either the answer or the rationale wrong, it receives no points.

Task setup The model is presented with a query $\boldsymbol{q}$, and four response choices $\boldsymbol{r}^{(i)}$. Like our model, we train the baselines using multi-class cross entropy between the set of

responses and the label. For simplicity, we train separate models for question answering and answer justification.[11]

7.1. Baselines

We compare our **R2C** to several strong language and vision baselines below:

Text-only baselines We evaluate the level of visual reasoning needed for the dataset by also evaluating purely text-only models. For each model, we represent $\boldsymbol{q}$ and $\boldsymbol{r}^{(i)}$ as streams of tokens, with the detection tags replaced by the object name (e.g. `chair5` is replaced with `chair`). To minimize the discrepancy between our task and pretrained models, we replace person detection tags with gender-neutral names.

a. **BERT** [17]: BERT is a recently released NLP model that achieves state-of-the-art performance on many NLP tasks.

b. **BERT (response only)** We use the same BERT model, however, during fine-tuning and testing the model is only given the response choices $\boldsymbol{r}^{(i)}$.

c. **ESIM+ELMo** [11]: ESIM is another high performing model for sentence-pair classification tasks, particularly when used with ELMo embeddings [58].

d. **LSTM+ELMo**: Here an LSTM with ELMo embeddings is used to score responses $\boldsymbol{r}^{(i)}$.

VQA Baselines Additionally we compare our approach to models developed on the VQA dataset [5]:

e. **RevisitedVQA** [39]: This model takes as input a query, response, and image features for the entire image, and passes the result through a multilayer perceptron, which has to classify 'yes' or 'no'.[12]

f. **Bottom-up and Top-down attention** (BottomUpTopDown) [4]: This model attends over region proposals given by a pretrained Faster-RCNN detector. To adapt to VCR, we pass this model objects referred to by the annotators.

g. **Multimodal Low-rank Bilinear Attention** (MLB) [43]: This model uses Hadamard products to merge the vision and language representations given by a query and each region in the image.

h. **Multimodal Tucker Fusion** (MUTAN) [6]: This model expresses joint vision-language context in terms of a tensor decomposition, allowing for more expressivity.

We note that BottomUpTopDown, MLB, and MUTAN all treat VQA as a multilabel classification over the top 1000 answers [4, 51]. Because VCR is highly diverse (Figure 4), for these models we represent each response $\boldsymbol{r}^{(i)}$ using a GRU [12].[13] The output logit for response i is given by the dot product between the final hidden state of the GRU encoding $\boldsymbol{r}^{(i)}$, and the final representation from the model.

[11] We follow the standard train, val and test splits.

[12] For VQA, the model is trained by sampling positive or negative answers for a given question; for our dataset, we simply use the result of the perceptron (for response $\boldsymbol{r}^{(i)}$) as the i-th logit.

[13] To match the other GRUs used in [4, 43, 6] which encode $\boldsymbol{q}$.

Model	$Q \rightarrow A$	$QA \rightarrow R$	$Q \rightarrow AR$
R2C	**63.8**	**67.2**	**43.1**
No query	48.3	43.5	21.5
No reasoning module	63.6	65.7	42.2
No vision representation	53.1	63.2	33.8
GloVe representations	46.4	38.3	18.3

Table 2: Ablations for **R2C**, over the validation set.

Human performance We asked five different workers on Amazon Mechanical Turk to answer 200 dataset questions from the test set. A different set of five workers were asked to choose rationales for those questions and answers. Predictions were combined using a majority vote.

7.2. Results and Ablations

We present our results in Table 1. Of note, standard VQA models struggle on our task. The best model, in terms of $Q \rightarrow AR$ accuracy, is MLB, with 17.2% accuracy. Deep text-only models perform much better: most notably, BERT [17] obtains 35.0% accuracy. One possible justification for this gap in performance is a bottlenecking effect: whereas VQA models are often built around multilabel classification of the top 1000 answers, VCR requires reasoning over two (often length) text spans. Our model, **R2C** obtains an additional boost over BERT by 9% accuracy, reaching a final performance of 44%. Still, this figure is nowhere near human performance: 85% on the combined task, so there is significant headroom remaining.

Ablations We evaluated our model under several ablations to determine which components are most important. Removing the query representation (and query-response contextualization entirely) results in a drop of 21.6% accuracy points in terms of $Q \rightarrow AR$ performance. Interestingly, this setting allows it to leverage its image representation more heavily: the text based response-only models (BERT response only, and LSTM+ELMo) perform barely better than chance. Taking the reasoning module lowers performance by 1.9%, which suggests that it is beneficial, but not critical for performance. The model suffers most when using GloVe representations instead of BERT: a loss of 24%. This suggests that strong textual representations are crucial to VCR performance.

Qualitative results Last, we present qualitative examples in Figure 7. **R2C** works well for many images: for instance, in the first row, it correctly infers that a bank robbery is happening. Moreover, it picks the right rationale: even though all of the options have something to do with 'banks' and 'robbery,' only **c)** makes sense. Similarly, analyzing the examples for which **R2C** chooses the right answer but the wrong rationale allows us to gain more insight into the model's understanding of the world. In the third row, the model incorrectly believes there is a crib while assigning

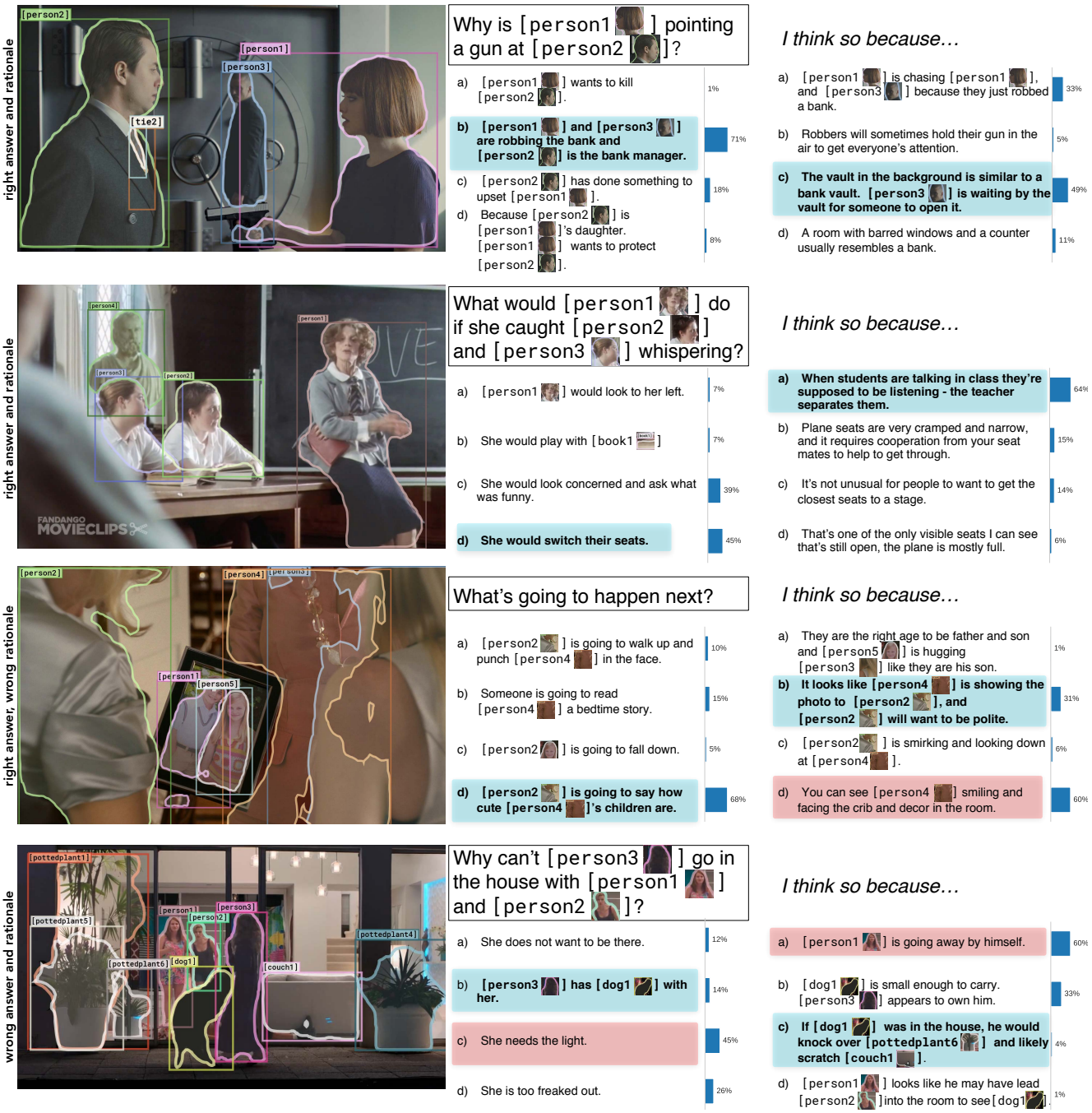


Figure 7: Qualitative examples from **R2C**. Correct choices are highlighted in **blue**, incorrect inferences are in red. The first two rows are successful cases, where the model correctly identifies the right answer and rationale - its probability judgments are to the right of each response choice. In the third row, the model chooses the right answer, but an incorrect rationale, referring to a 'crib' that is not in the room. In the last row, the model chooses the wrong answer and rationale.

less probability mass on the correct answer - that everyone is looking at a photo of children, not actual children.

Acknowledgements

We thank the Mechanical Turk workers for doing such an outstanding job with dataset creation - this dataset and paper would not exist without them. Thanks also to Michael Schmitz for helping with the dataset split and Jen Dumas for legal advice, along with the suggestion to use MovieClips. This work was supported by the National Science Foundation Graduate Research Fellowship (DGE-1256082), the NSF grant (IIS-1524371, 1703166), the DARPA CwC program through ARO (W911NF-15-1-0543), the IARPA DIVA program through D17PC00343, and gifts by Google and Facebook. The views and conclusions contained herein are those of the authors and should not be interpreted as representing endorsements of IARPA, DOI/IBC, or the U.S. Government.

Appendix

Abstract

*In our work we presented the new task of Visual Commonsense Reasoning and introduced a large-scale dataset for the task, VCR, along with **Adversarial Matching**, the machinery that made the dataset construction possible. We also presented **R2C**, a new model for the task. In the supplemental material, we provide the following items that shed further insight on these contributions:*

- *Additional dataset analysis (Section A)*
- *More information about dataset creation (Section B) and **Adversarial Matching** (Section C)*
- *An extended discussion on language priors (Section D)*
- *Model hyperparameters used (Section E)*
- *A visualization of **R2C**'s predictions (Section F)*

For more examples, and to obtain the dataset and code, check out visualcommonsense.com.

A. Dataset Analysis

In this section, we continue our high-level analysis of VCR. Some statistics are shown in Table 3. These statistics suggest a challenging and diverse dataset: unlike many question-answering datasets wherein the answer is a single word, our answers average to more than 7.5 words. The rationales are even longer, averaging at more than 16 words.

On average, there are roughly two objects mentioned over a question, answer, and rationale. Most of these objects are people (Figure 8), though other types of COCO objects are common too [50]. Objects such as 'chair,' 'tie,' and 'cup' are often detected, however, these objects vary in terms of scene importance: even though more ties exist in the data than cars, workers refer to cars more in their questions, answers, and rationales. Some objects, such as hair driers and snowboards, are rarely detected.

Our dataset also covers a broad range of movies - over 2000 in all, mostly via MovieClips (Figure 9). We note that since we split the dataset by movie, the validation and test sets cover a completely disjoint set of movies, which forces a model to generalize. For each movie image, workers ask 2.6 questions on average (Figure 10), though the exact number varies - by design, workers ask more questions for more interesting images.

Still, it is difficult to accurately estimate the level of commonsense and cognition-level phenomena in the dataset. One approach that we presented in the paper was to categorize questions by type: to estimate this over the entire training set, we used a several patterns, which we show in Table 4. Still, we note that automatic categorization of the inference types required for this task is hard. This is in part because a single question might require multiple types of reasoning: for example, 'Why does person1 feel embarrassed?' requires reasoning about person1's mental state, as well as requiring an explanation. For this reason, we argue that this breakdown underestimates the task difficulty.

	Train	Val	Test
Number of questions	212,923	26,534	25,263
Number of answers per question	4	4	4
Number of rationales per question	4	4	4
Number of images	80,418	9,929	9,557
Number of movies covered	1,945	244	189
Average question length	6.61	6.63	6.58
Average answer length	7.54	7.65	7.55
Average rationale length	16.16	16.19	16.07
Average # of objects mentioned	1.84	1.85	1.82

Table 3: High level dataset statistics, split by fold (train, validation, and test). Note that we held out one fold in the dataset for blind evaluation at a later date; this fold is blind to us to preserve the integrity of the held-out data. Accordingly, the statistics of that fold are not represented here.

Type	Freq.	Patterns
Explanation	38%	why, how come, how does
Activity	24%	doing, looking, event, playing, preparing
Temporal	13%	happened, before, after, earlier, later, next
Mental	8%	feeling, thinking, saying, love, upset, angry
Role	7%	relation, occupation, strangers, married
Scene	5%	where, time, near
Hypothetical	5%	if, would, could, chance, might, may

Table 4: Some of the rules we used to determine the type of each question. Any question containing a word from one of the above groups (such as 'why') was determined to be of that type ('explanation').

B. Dataset Creation Details

In this section, we elaborate more on how we collected VCR, and about our crowdsourcing process.

B.1. Shot detection pipeline

The images in VCR are extracted from video clips from LSMDC [68] and MovieClips. These clips vary in length from a few seconds (LSMDC) to several minutes (MovieClips). Thus, to obtain more still images from these clips, we performed shot detection. Our pipeline is as follows:

- We iterate through a video clip at a speed of one frame per second.
- During iteration, we also perform shot detection: if we

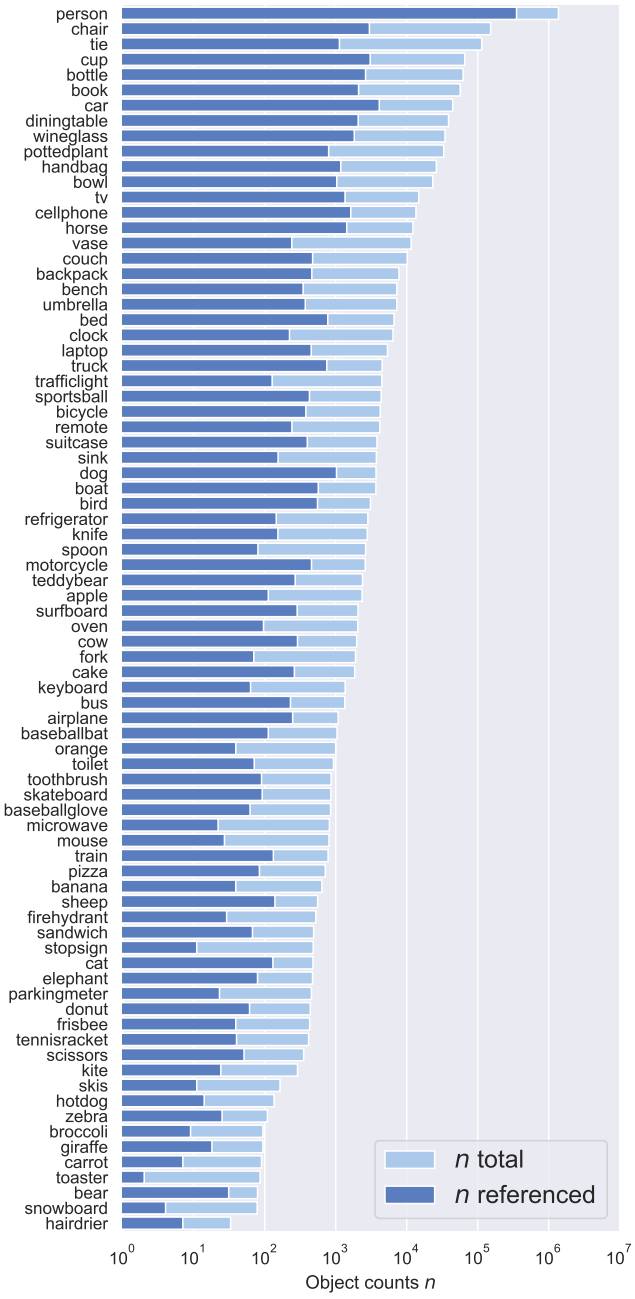


Figure 8: Distribution of the referenced COCO [50] objects in VCR. We count an object as being 'referenced' if, for a given question, answer, and rationale, that object is mentioned explicitly. Note that we do not double-count objects here - if `person5` is mentioned in the question and the answer, we count it once. This chart suggests that our dataset is mostly human-centric, with some categories being referenced more than others (cars are mentioned more than ties, even though cars appear less often).

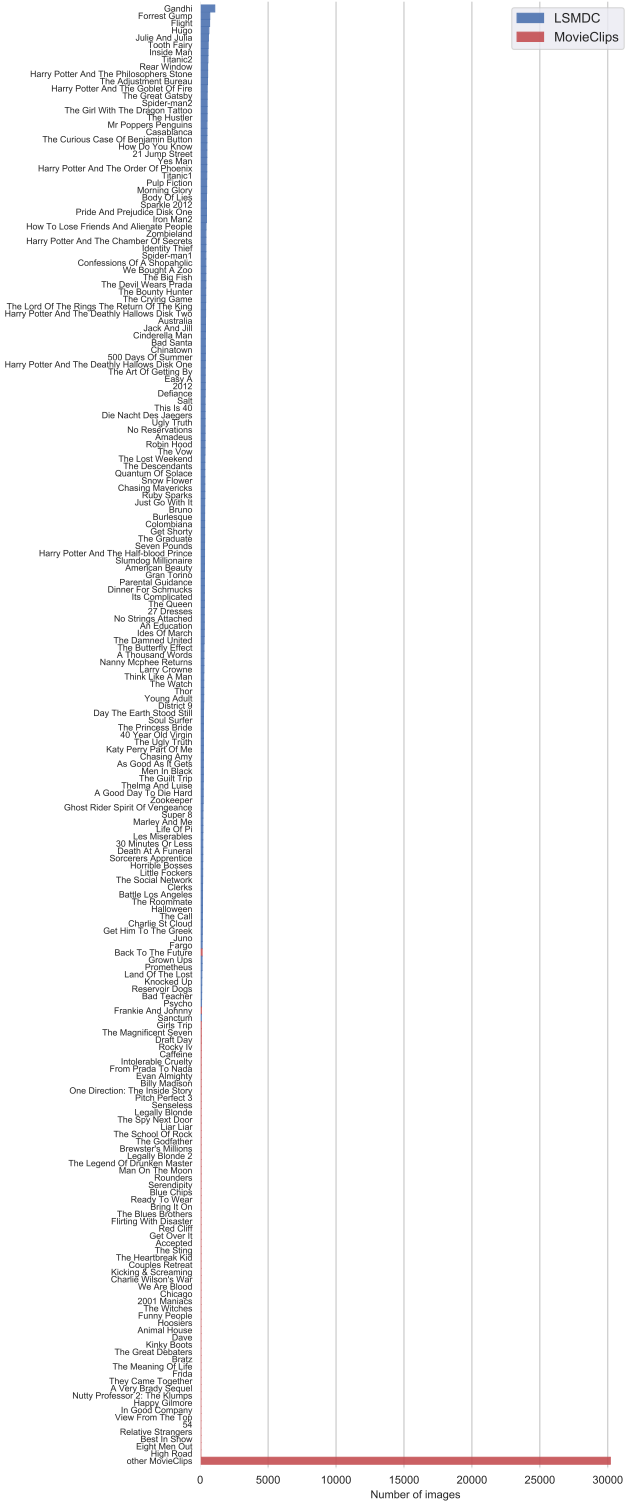


Figure 9: Distribution of movies in the VCR training set by number of images. Blue bars are movies from LSMDC (46k images); red are MovieClips (33k images). The MovieClips images are spread over a wider range of movies: due to space restrictions, most are under 'other MovieClips.'

detect a mean difference of 30 pixels in HSV space, then we register a shot boundary.

- After a shot boundary is found, we apply Mask-RCNN

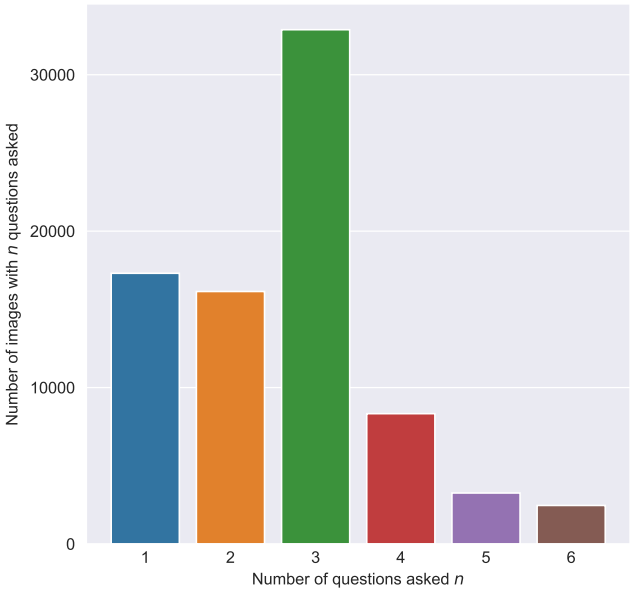
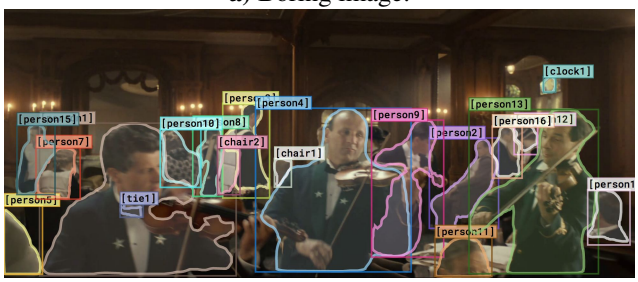


Figure 10: Number of questions asked per image on the VCR training set. The average number of questions asked per image is 2.645. Note that while workers could ask anywhere between one to three questions per image, images that were flagged as especially interesting by workers got re-annotated with additional annotations.

a) Boring image.



b) Interesting image.

Figure 11: Two example images that come from the raw video pipeline. Image a) is flagged by our initial filter as 'boring', because there are only two people without any additional objects, whereas image b) is flagged as being interesting due to the number of people and objects detected.

[30, 25] on the middle frame for the shot, and save the resulting image and detection information.

We used a threshold of 0.7 for Mask-RCNN, and the best detection/segmentation model available for us at the time: X-101-64x4d-FPN[14], which obtains 42.4 box mAP on COCO, and 37.5 mask mAP.

B.2. Interestingness Filter

Recall that we use an 'interestingness filter' to ensure that the images in our dataset are high quality. First, every image had to have at least two people in it, as detected by Mask RCNN. However, we also found that many images with two or more people were still not very interesting. The two main failure cases here are when there are one or two people detected, but they aren't doing anything interesting (Figure 11a), or when the image is especially grainy and blurry. Thus, we opted to learn an additional classifier for determining which images were interesting.

Our filtering process evolved as we collected data for the task. The first author of this paper first manually annotated 2000 images from LSMDC [68] as being 'interesting' or 'not interesting' and trained a logistic regression model to predict said label. The model is given as input the number of people detected by Mask RCNN [30, 25], along with the number of objects (that are not people) detected. We used this model to identify interesting images in LSMDC, using a threshold that corresponded to 70% precision. This resulted in 72k images selected; these images were annotated first.

During the crowdsourcing process, we obtained data that allowed us to build an even better interesting filter later on. Workers were asked, along with each image, whether they thought that the image was especially interesting (and thus should go to more workers), just okay, or especially boring (and hard to ask even one good question for). We used this to train a deeper model for this task. We used the Resnet50 architecture as our backbone [31]. The model uses a Resnet 50 backbone over the entire image [31] as well as a multilayer perceptron over the object counts. The entire model is trained end-to-end: 2048 dimensional features from Resnet are concatenated with a 512 dimensional projetion of the object counts, and used to predict the labels.[15] We used this model to select the most interesting 40k images from Movieclips, which finished off the annotation process.

B.3. Crowdsourcing quality data

As mentioned in the paper, crowdsourcing data at the quality and scale of VCR is challenging. We used several best practices for crowdsourcing, which we elaborate on in this section.

[14] Available via the Detectron Model Zoo.

[15] In addition to predicting interestingness, the model also predicts the number of questions a worker asks, but we never ended up using these predictions.

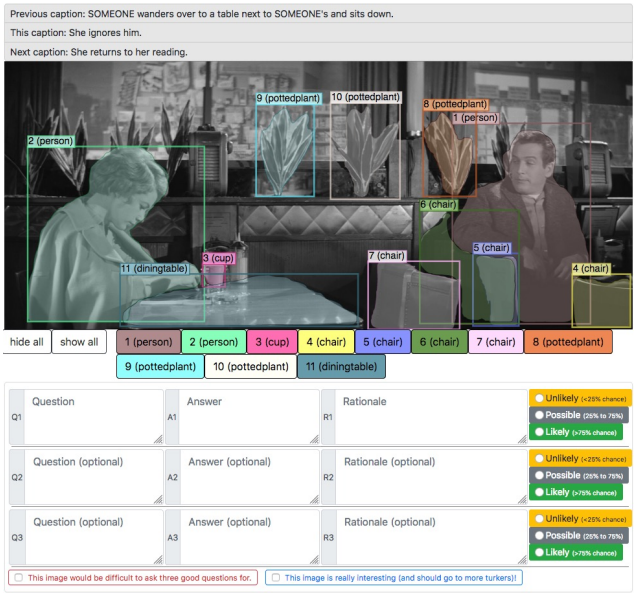


Figure 12: Screenshot of our annotation interface. Workers are given an image, as well as context from the video (here, captions from LSMDC [68]), and are asked to write one to three questions, answers, and rationales. For each answer, they must mark it as likely, possible, or unlikely. Workers also select whether the image was especially interesting or boring, as this allows us to train a deep model for predicting image interestingness.

We used Amazon Mechanical Turk for our crowdsourcing. A screenshot of our interface is given in Figure 12. Given an image, workers asked questions, answered them, and provided a rationale explaining why their answer might be correct. These are all written in a mixture of natural language text, as well as referring to detection regions. In our annotation UI, workers refer to the regions by writing the tag number.[16]

Workers could ask anywhere between one to three questions per HIT. We paid the workers proportionally at $0.22 per triplet. According to workers, this resulted in $8–25/hr. This proved necessary as workers reported feeling "drained" by the high quality required.

Automated quality checks We added several automated checks to the crowdsourcing UI to ensure high quality. The workers had to write at least four words for the question, three for the answer, and five for the rationale. Additionally, the workers had to explicitly refer to at least one detection on average per question, answer, and rationale triplet. This was automatically detected to ensure that the workers were referring to the detection tags in their submissions.

We also noticed early on was that sometimes workers would write detailed stories that were only loosely connected with the semantic content of the image. To fix this, workers also had to self-report whether their answer was likely (above 75% probability), possible (25-75% probability), or unlikely (below 25% probability). We found that this helped deter workers from coming up with consistently unlikely answers for each image.

Instructions Like for any crowdsourcing task, we found wording the instructions carefully to be crucial. We encouraged workers to ask about higher-level actions, versus lower-level ones (such as 'What is `person1` wearing?'), as well as to not ask questions and answers that were overly generic (and thus could apply to many images). Workers were encouraged to answer reasonably in a way that was not overly unlikely or unreasonable. To this end, we provided the workers with high-quality example questions, answers, and rationales.

Qualification exam Since we were picky about the types of questions asked, and the format of the answers and rationales, workers had to pass a qualification task to double check that they understood the format. The qualification test included a mix of multiple-choice graded answers as well as a short written section, which was to provide a single question, answer, and rationale for an image. The written answer was checked manually by the first author of this paper.

Work verification In addition to the initial qualification exam, we also periodically monitored the annotation quality. Every 48 hours, the first author of this paper would review work and provide aggregate feedback to ensure that workers were asking good questions, answering them well, and structuring the rationales in the right way. Because this took significant time, we then selected several outstanding workers and paid them to do this job for us: through a separate set of HITs, these outstanding workers were paid $0.40 to provide detailed feedback on a submission that another worker made. Roughly one in fifty HITs were annotated in this way to give extra feedback. Throughout this process, workers whose submission quality dropped were dequalified from the HITs.

C. Adversarial Matching Details

There are a few more details that we found useful when performing the **Adversarial Matching** to create VCR, which we discuss in this section.

Aligning Detections In practice, most responses in our dataset are not relevant to most questions, due to the diversity of responses in our dataset and the range of detection tags (`person1`, etc.).

To fix this, for each query $\boldsymbol{q}_i$ (with associated object list $\boldsymbol{o}_i$ and response $\boldsymbol{r}_i$) we turn each candidate $\boldsymbol{r}_j$ into a tem-

[16] Note that this differs a bit from the format in the paper: we originally had workers write out the full tag, like `[person5]`, but this is often long and the workers would sometimes forget the brackets. Thus, the tag format here is just a single number, like `5`.

plate, and use a rule based system to probabilistically remap its detection tags to match the objects in $\boldsymbol{o}_i$. With some probability, a tag in $\boldsymbol{r}_j$ is replaced with a tag in $\boldsymbol{q}_i$ and $\boldsymbol{r}_i$. Otherwise, it is replaced with a random tag from $\boldsymbol{o}_i$.

We note that our approach isn't perfect. The remapping system often produces responses that violate predicate/argument structure, such as '`person1` is kissing `person1`.' However, *our approach does not need to be perfect*: because the detections for response $\boldsymbol{r}_j$ are remapped uniquely for each query $\boldsymbol{q}_i$, with some probability, there should be at least some remappings of $\boldsymbol{r}_i$ that make sense, and the question relevance model P_{rel} should select them.

Semantic categories Recall that we use 11 folds for the dataset of around 290k questions, answers, and rationales. Since we must perform **Adversarial Matching** once for the answers, as well as for the rationales, this would naively involve 22 matchings on a fold size of roughly 26k. We found that the major computational bottleneck wasn't the bipartite matching[17], but rather the computation of all-pairs similarity and relevance between $\sim$26k examples.

There is one additional potential problem: we want the dataset examples to require a lot of complex commonsense reasoning, rather than simple attribute identification. However, if the response and the query disagree in terms of gender pronouns, then many of the dataset examples can be reduced to gender identification.

We address both of these problems by dividing each fold into 'buckets' of 3k examples for matching. We divide the examples up in terms of the pronouns in the response: if the response contains a female or male pronoun, then we put the example into a 'female' or 'male' bucket, respectively, otherwise the response goes into the 'neutral' bucket. To further divide the dataset examples, we also put different question types in different buckets for the question answering task (e.g. who, what, etc.). For the answer justification task, we cluster the questions and answers using their average GloVe embeddings [57].

Relevance model details Recall that our relevance model P_{rel} is trained to predict the probability that a response $\boldsymbol{r}$ is valid for a query $\boldsymbol{q}$. We used BERT for this task [17], as it achieves state-of-the-art results across many two-sentence inference tasks. Each input looks like the following, where the query and response are concatenated with a separator in between:

```
[CLS] what is casey doing ?  [SEP] casey
is getting out of car .  [SEP]
```

Note that in the above example, object tags are replaced with the class name (`car3`→`car`). Person tags are replaced with gender neutral names (`person1`→`casey`) [21].

[17] We use the https://github.com/gatagat/lap implementation.

We finetune BERT by treating it as a two-way classification problem. With probability 25% for a query, BERT is given that query's actual response, otherwise it is given a random response (where the detections were remapped). Then, the model must predict whether it was given the actual response or not. We used a learning rate of $2 \cdot 10^{-5}$, the Adam optimizer [45], a batch size of 32, and 3 epochs of finetuning.[18]

Due to computational limitations, we used BERT-Base as the architecture rather than BERT-Large - the latter is significantly slower.[19] Already, P_{rel} has an immense computational requirement as it must compute all-pairs similarity for the entire dataset, over buckets of 3000 examples. Thus, we opted to use a large bucket size rather than a more expensive model.

Similarity model details While we want the responses to be highly relevant to the query, we also want to avoid cases where two responses might be conflated by humans - particularly when one is the correct response. This conflation might occur for several reasons: possibly, two responses are *paraphrases* of one another, or one response *entails* another. We lump both under the 'similarity' umbrella as mentioned in the paper and introduce a model, P_{sim}, to predict the probability of this occurring - broadly speaking, that two responses $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ have the same meaning.

We used ESIM+ELMo for this task [11, 58], as it still does quite well on two-sentence natural language inference tasks (although not as well as BERT), and can be made much more efficient. At test time, the model makes the similarity prediction when given two token sequences.[20]

We trained this model on freely available NLP corpora. We used the SNLI formalism [8], in which two sentences are an 'entailment' if the first entails the second, 'contradiction' if the first is contradicted by the second, and 'neutral' otherwise. We combined data from SNLI and MultiNLI [83] as training data. Additionally, we found that even after training on these corpora, the model would struggle with paraphrases, so we also translated SNLI sentences from English to German and back using the Nematus machine translation system [82, 74]. These sentences served as extra paraphrase data and were assigned the 'entailment' label. We also used randomly sampled sentence pairs from SNLI as additional 'neutral' training data. We held out the SNLI validation set to determine when to stop training. We used standard hyperparameters for ESIM+ELMo as given by the AllenNLP library [24].

[18] We note that during the **Adversarial Matching** process, for either Question Answering or Answer Justification, the dataset is broken up into 11 folds. For each fold, BERT is finetuned on the other folds, not on the final dataset splits.

[19] Also, BERT-Large requires much more memory, enough so that it's harder to finetune due to the smaller feasible batch size.

[20] Again, with object tags replaced with the class name, and person tags replaced by gender neutral names.

Given the trained model P_{nli}, we defined the similarity model as the maximum entailment probability for either way of ordering the two responses:

$$P_{sim}(\boldsymbol{r}_i, \boldsymbol{r}_j) = \max\Big\{P_{nli}(\text{ent}|\boldsymbol{r}_i, \boldsymbol{r}_j), P_{nli}(\text{ent}|\boldsymbol{r}_j, \boldsymbol{r}_i)\Big\}, \tag{3}$$

where 'ent' refers to the 'entailment' label. If one response entails the other, we flag them as similar, even if the reverse entailment is not true, because such a response is likely to be a false positive as a distractor.

The benefit of using ESIM+ELMo for this task is that it can be optimized for the task of all-pair sentence similarity. While much of the ESIM architecture involves computing attention between the two text sequences, everything before the first attention can be precomputed. This provides a large speedup, particularly as computing the ELMo representations is expensive. Now, for a fold size of N, we only have to compute $2N$ ELMo representations rather than N^2.

Validating the λ parameter Recall that our hyperparameter λ trades off between machine and human difficulty for our final dataset. We shed more insight on how we chose the exact value for λ in Figure 13. We tried several different values of λ and chose $\lambda = 0.1$ for $Q \rightarrow A$ and $\lambda = 0.01$ for $QA \rightarrow R$, as at these thresholds human performance was roughly 90%. For an easier dataset for both humans and machines, we would increase the hyperparameter.

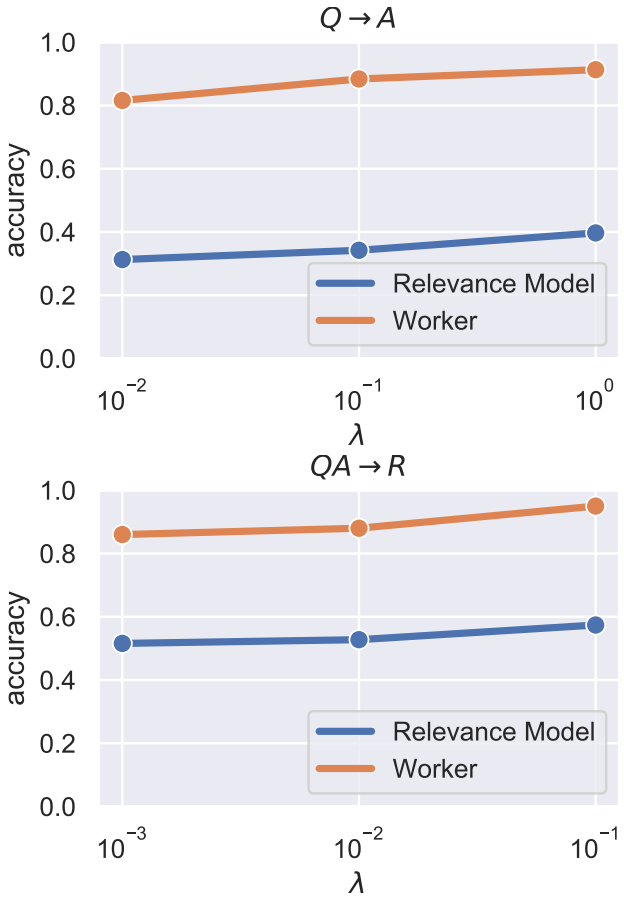


Figure 13: Tuning the λ hyperparameter. Workers were asked to solve 100 dataset examples from the validation set, as given by **Adversarial Matching** for each considered value of λ. We used these results to pick reasonable values for the hyperparameter such that the task was difficult for the question relevance model P_{rel}, while simple for human workers. We chose $\lambda = 0.1$ for $Q \rightarrow A$ and $\lambda = 0.01$ for $QA \rightarrow R$.

D. Language Priors and Annotation Artifacts Discussion

There has been much research in the last few years in understanding what 'priors' datasets have.[21] Broadly speaking, how well do models do on VCR, as well as other visual question answering tasks, without vision?

To be more general, we will consider problems where a model is given a *question* and *answer choices*, and picks exactly one answer. The *answer choices* are the outputs that the model is deciding between (like the responses in VCR) and the *question* is the shared input that is common to all *answer choices* (the query, image, and detected objects in VCR). With this terminology, we can categorize unwanted dataset priors in the following ways:

- **Answer Priors**: A model can select a correct answer without even looking at the question. Many text-only datasets contain these priors. For instance, the RocStories dataset [54] (in which a model must classify endings to a story as correct or incorrect), a model can obtain 75% accuracy by looking at stylistic features (such as word choice and punctuation) in the endings.
- **Non-Visual Priors**: A model can select a correct answer using only non-visual elements of the question. One example is VQA 1.0 [5]: given a question like 'What color is the fire hydrant?' a model will classify some answers higher than others (red). This was addressed in VQA 2.0 [27], however, some answers will still be more likely than others (VQA's answers are open-ended, and an answer to 'What color is the fire hydrant?' must be a color).

These priors can either arise from biases in the world (fire hydrants are usually red), or, they can come from annotation artifacts [29]: patterns that arise when people write class-conditioned answers. Sometimes these biases are subliminal: when asked to write a correct or incorrect story ending, the correct endings tend to be longer [73]. Other

[21] This line of work is complementary to other notions of dataset bias, like understanding what phenomena datasets cover or don't [77], particularly how that relates to how marginalized groups are represented and portrayed [72, 91, 70, 69].

cases are more obvious: workers often use patterns such as negation to write sentences that contradict a sentence [29].[22]

To what extent do vision datasets suffer from annotation artifacts, versus world priors? We narrow our focus to multiple-choice question answering datasets, in which for humans traditionally write correct *and* incorrect answers to a question (thus, potentially introducing the annotation artifacts). In Table 5 we consider several of these datasets: TVQA [47], containing video clips from TV shows, along with subtitles; MovieQA [76], with videos from movies and questions obtained from higher-level plot summaries; PororoQA [44], with cartoon videos; and TGIFQA [40], with templated questions from the TGIF dataset [48]. We note that these all differ from our proposed VCR in terms of subject matter, questions asked, number of answers (each of the above has 5 answers possible, while we have 4) and format; our focus here is to investigate how difficult these datasets are for text-only models.[23] Our point of comparison is VCR, since our use of **Adversarial Matching** means that humans never write incorrect answers.

We tackle this problem by running BERT-Base on these models [17]: given only the answer (A), the answer and the question (Q+A), or additional language context in the form of subtitles (S+Q+A), how well does BERT do? Our results in Table 5 help support our hypothesis regarding annotation artifacts: whereas accuracy on VCR, only given the ending, is 27% for $Q \rightarrow A$ and 26% for $Q \rightarrow A$, versus a 25% random baseline. Other models, where humans write the incorrect answers, have answer-only accuracies from 33.8% (MovieQA) to 45.8% (TGIFQA), over a 20% baseline.

There is also some non-visual bias for all datasets considered: from 35.4% when given the question and the answers (MovieQA) to 72.5% (TGIFQA). While these results suggest that MovieQA is incredibly difficult without seeing the video clip, there are two things to consider here. First, MovieQA is roughly 20x smaller than our dataset, with 9.8k examples in training. Thus, we also tried training BERT on 'VCR$^{\text{small}}$': taking 9.8k examples at random from our training set. Performance is roughly 14% worse, to the point of being roughly comparable to MovieQA.[24] Second, often times the examples in MovieQA have similar structure, which might help to alleviate stylistic priors, for example:

"Who has followed Boyle to Eamon's apartment?" Answers:

1. Thommo and his IRA squad.
2. Darren and his IRE squad.
3. Gary and his allies.
4. **Quinn and his IRA squad.**
5. Jimmy and his friends.

On the other hand, our dataset examples tend to be highly diverse in terms of syntax as well as high-level meaning, due to the similarity penalty. We hypothesize that this is why some language priors creep into VCR, particularly in the $QA \rightarrow R$ setting: given four very distinct rationales that ostensibly justify why an answer is true, some will likely serve as better justifications than others.

Furthermore, providing additional language information (such as subtitles) to a model tends to boost performance considerably. When given access to subtitles in TVQA,[25]

Dataset	$\#_{train}$	Chance	A	Q+A	S+Q+A
TVQA [47]	122,039	20.0	45.0	47.4	70.6♠
MovieQA [76]	9,848	20.0	33.8	35.4	36.3♣
PororoQA [44]♡	7,530	20.0	43.1	47.4	
TGIFQA [40]♢	73,179	20.0	45.8	72.5	
VCR $Q \rightarrow A$	212,923	25.0	27.6	53.8	
VCR $QA \rightarrow R$		25.0	26.3	64.1	
VCR$^{\text{small}}$ $Q \rightarrow A$	9,848	25.0	25.5	39.9	
VCR$^{\text{small}}$ $QA \rightarrow R$		25.0	25.3	50.9	

Table 5: Text-only results on the validation sets of vision datasets, using BERT-Base. $\#_{train}$ shows the number of training examples. A corresponds to only seeing the answer; in Q+A the model also sees the question; in S+Q+A the model also sees subtitles from the video clip. These results suggest that many multiple choice QA datasets suffer from annotation artifacts, while **Adversarial Matching** helps produce a dataset with minimial biases; morover, providing extra text-only information (like subtitles) greatly boosts performance. More info:

♠: State of the art.

♣: Only 45% (879/1958) of the questions in the MovieQA validation set have timestamps, which are needed to extract clip-level subtitles, so for the other 55%, we don't use any subtitle information.

♡: No official train/val/test split is available, so we split the data by movie, using 20% of data for validation and the rest for training.

♢: There seem to be issues with the publicly released train-test split of TGIFQA (namely, a model with high accuracy on a held-out part of the training set doesn't generalize to the provided test set) so we resplit the multiple-choice data ourselves by GIF and hold out 20% for validation.

[22] For instance, the SNLI dataset contains pairs of sentences with labels such as 'entailed' or 'contradiction' [8]. For a sentence like 'A skateboarder is doing tricks' workers often write 'Nobody is doing tricks' which is a contradiction. The result is that the word 'nobody' is highly predictive of a word being a contradiction.

[23] It should be noted that all of these datasets were released before the existence of strong text-only baselines such as BERT.

[24] Assuming an equal chance of choosing each incorrect ending, the results for BERT on an imaginary 4-answer version of TVQA and MovieQA would be 54.5% and 42.2%, respectively.

[25] We prepend all of the subtitles that are aligned to the video clip to the beginning of the question, with a special token in between (';'). We cut off tokens from the subtitles when the total sequence length is above 128

BERT scores 70.6%, which to the best of our knowledge is a new state-of-the-art on TVQA.

In conclusion, dataset creation is highly difficult, particularly as there are many ways that unwanted bias can creep in during the dataset creation process. One such bias of this form includes annotation artifacts, which our analysis suggests is prevalent amongst multiple-choice VQA tasks wherein humans write the wrong endings. Our analysis also suggests **Adversarial Matching**, can help minimize this effect, even when there are strong natural biases in the underlying textual data.

E. Model details

In this section, we discuss implementation details for our model, **R2C**.

BERT representations As mentioned in the paper, we used BERT to represent text [17]. We wanted to provide a fair comparison between our model and BERT, so we used BERT-Base for each.[26]

We tried to make our use of BERT to be as simple as possible, matching our use of it as a baseline. Given a query $\boldsymbol{q}$ and response choice $\boldsymbol{r}^{(i)}$, we merge both into a single sequence to give to BERT. One example might look like the following:

```
[CLS] why is riley riding motorcycle
while wearing a hospital gown ? [SEP]
she had to leave the hospital in a hurry
. [SEP]
```

Note that in the above example, we replaced person tags with gender neutral names [21] (`person3`→ `riley`) and replaced object detections by their class name (`motorcycle1`→ `motorcycle`), to minimize domain shift between BERT's pretrained data (Wikipedia and the BookCorpus [95]) and VCR.

Each token in the sequence corresponds to a different transformer unit in BERT. We can then use the later layers in BERT to extract contextualized representations for the each token in the query (everything from `why` to `?`) and the response (`she` to `.`).[27] Note that this gives us a different representation for each response choice i.

We extract frozen BERT representations from the second-to-last layer of the transformer.[28] Intuitively, this makes sense as the representations that that layer are used for both of BERT's pretraining tasks: next sentence prediction (the unit corresponding to the `[CLS]` token at the last layer L attends to all units at layer $L-1$), as well as masked language modeling (the unit for a word at layer L looks at its hidden state at the previous layer $L-1$, and uses that to attend to all other units as well). The experiments in [17] suggest that this works well, though not as well as finetuning BERT end-to-end or concatenating multiple layers of activations.[29] The tradeoff, however, is that precomputing BERT representations lets us substantially reduce the runtime of **R2C** and allows us to focus on learning more powerful vision representations.

Model Hyperparameters A more detailed discussion of the hyperparameters used for **R2C** is as follows. We tried to stick to simple settings (and when possible, used similar configurations for the baselines, particularly with respect to learning rates and hidden state sizes).

- Our projection of image features maps a 2176 dimensional hidden size (2048 from ResNet50 as well as 128 dimensional class embeddings) to a 512 dimensional vector.
- Our grounding LSTM is a single-layer bidirectional LSTM with a 1280-dimensional input size (768 from BERT and 512 from image features) and uses 256 dimensional hidden states.
- Our reasoning LSTM is a two-layer bidirectional LSTM with a 1536-dimensional input size (512 from image features, and 256 for each direction in the attended, grounded query and the grounded answer). It also uses 256-dimensional hidden states.
- The representation from the reasoning LSTM, grounded answer, and attended question is maxpooled and projected to a 1024-dimensional vector. That vector is used to predict the ith logit.
- For all LSTMs, we initialized the hidden-hidden weights using orthogonal initialization [71], and applied recurrent dropout to the LSTM input with $p_{drop} = 0.3$ [23].
- The Resnet50 backbone was pretrained on Imagenet [16, 31]. The parameters in the first three blocks of ResNet were frozen. The final block (after the RoiAlign is applied) is finetuned by our model. We were worried, however, that the these representations would drift and so we added an auxiliary loss to the model inspired by [49]: the 2048-dimensional representation of each object (without class embeddings) had to be predictive of that object's label (via a linear projection to the label space and a softmax).

tokens.

[26] As mentioned elsewhere, we don't have access to TPUs at Google scale, making it hard to finetune BERT-Large. At a low batch size, the finetuning process has been reported to be unstable.

[27] The only slight difference is that, due to the WordPiece encoding scheme, rare words (like `chortled`) are broken up into subword units (`cho ##rt ##led`). In this case, we represent that word as the average of the BERT activations of its subwords.

[28] Since the domain that BERT was pretrained on (Wikipedia and the BookCorpus [95]) is still quite different from our domain, we finetuned BERT on the text of VCR (using the masked language modeling objective, as well as next sentence prediction) for one epoch to account for the domain shift, and then extracted the representations.

[29] This suggests, however, that if we also finetuned BERT along with the rest of the model parameters, the results of **R2C** would be higher.

- Often times, there are a lot of objects in the image that are not referred to by the query or response set. We filtered the objects considered by the model to include only the objects mentioned in the query and responses. We also passed in the entire image as an 'object' that the model could attend to in the object contextualization layer.
- We optimized **R2C** using Adam [45], with a learning rate of $2 \cdot 10^{-4}$ and weight decay of 10^{-4}. Our batch size was 96. We clipped the gradients to have a total L_2 norm of at most 1.0. We lowered the learning rate by a factor of 2 when we noticed a plateau (validation accuracy not increasing for two epochs in a row). Each model was trained for 20 epochs, which took roughly 20 hours over 3 NVIDIA Titan X GPUs.

F. Additional qualitative results

In this section, we present additional qualitative results from **R2C**. Our use of attention mechanisms allow us to better gain insight into how the model arrives at its decisions. In particular, the model uses the answer to attend over the question, and it uses the answer to attend over relevant objects in the image. Looking at the attention maps help to visualize which items in the question are important (usually, the model focuses on the second half of the question, like 'covering his face' in Figure 14), as well as which objects are important (usually, the objects referred to by the answer are assigned the most weight).

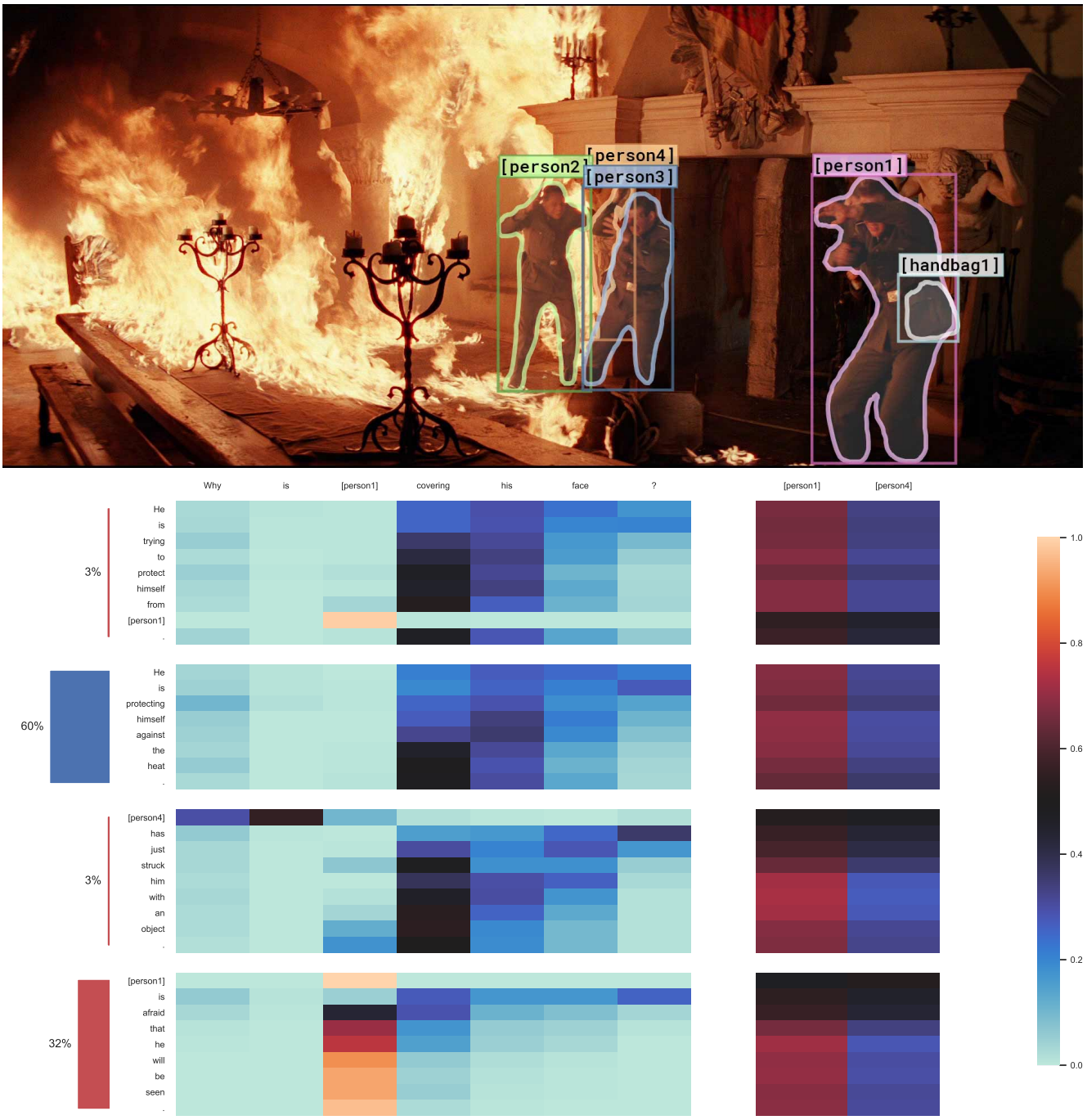


Figure 14: An example from the $Q \rightarrow A$ task. Each super-row is a response choice (four in total). The first super-column is the question: Here, 'Why is `[person1] covering his face?`' and the second super-column represents the relevant objects in the image that **R2C** attends to. Accordingly, each block is a heatmap of the attention between each response choice and the query, as well as each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 60% confident that the right answer is **b.**, which is correct.

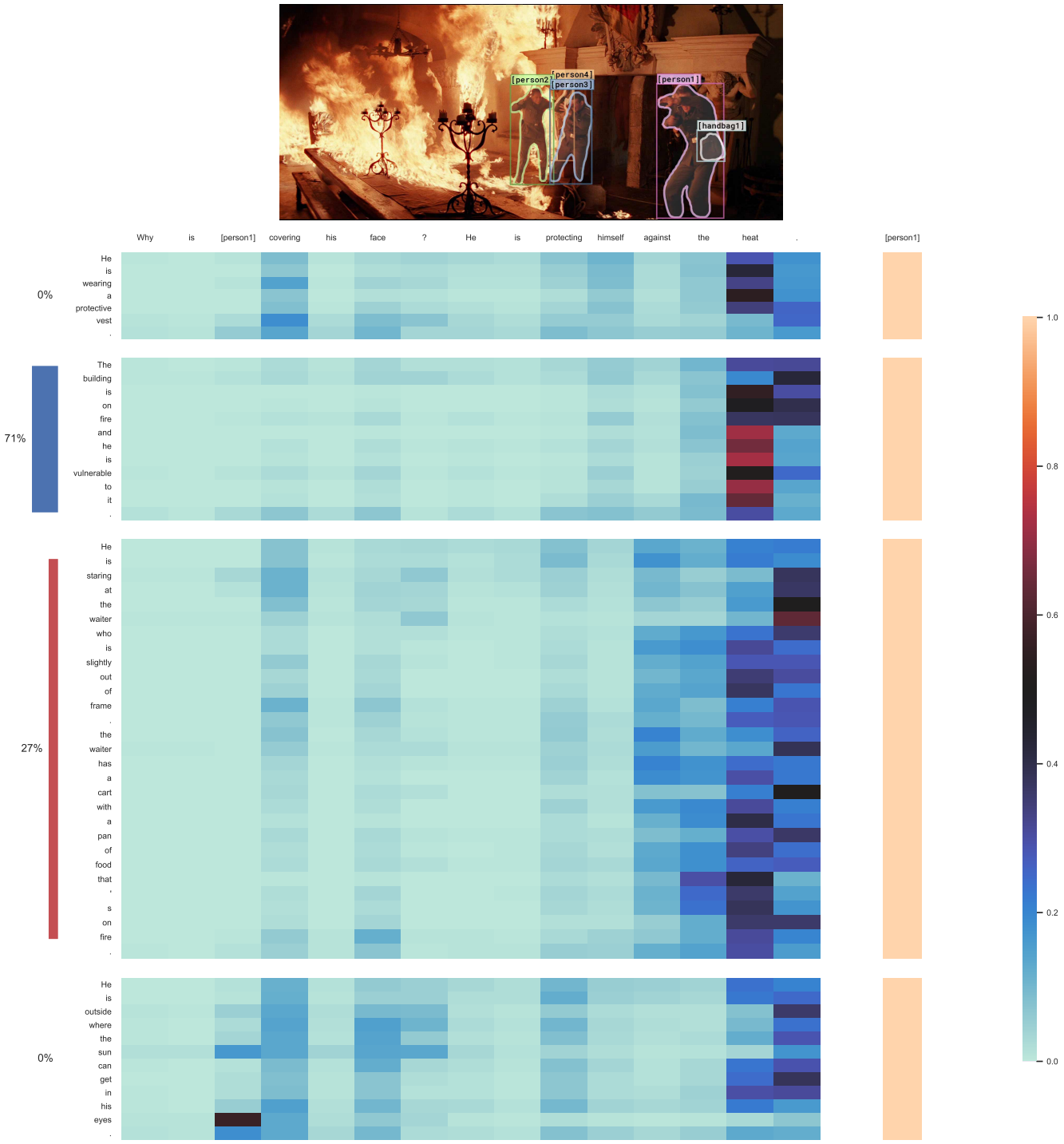


Figure 15: An example from the $QA \rightarrow R$ task. Each super-row is a response choice (four in total). The first super-column is the query, and the second super-column holds the relevant objects (here just a single person, as no other objects were mentioned by the query or responses). Each block is a heatmap of the attention between each response choice and the query, as well as the attention between each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 71% confident that the right answer is **b.**, which is correct.

Figure 16: An example from the $Q \rightarrow A$ task. Each super-row is a response choice (four in total). The first super-column is the question: Here, 'What is `[person13] doing?`' and the second super-column represents the relevant objects in the image that **R2C** attends to. Accordingly, each block is a heatmap of the attention between each response choice and the query, as well as each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 86% confident that the right answer is **d.**, which is correct.

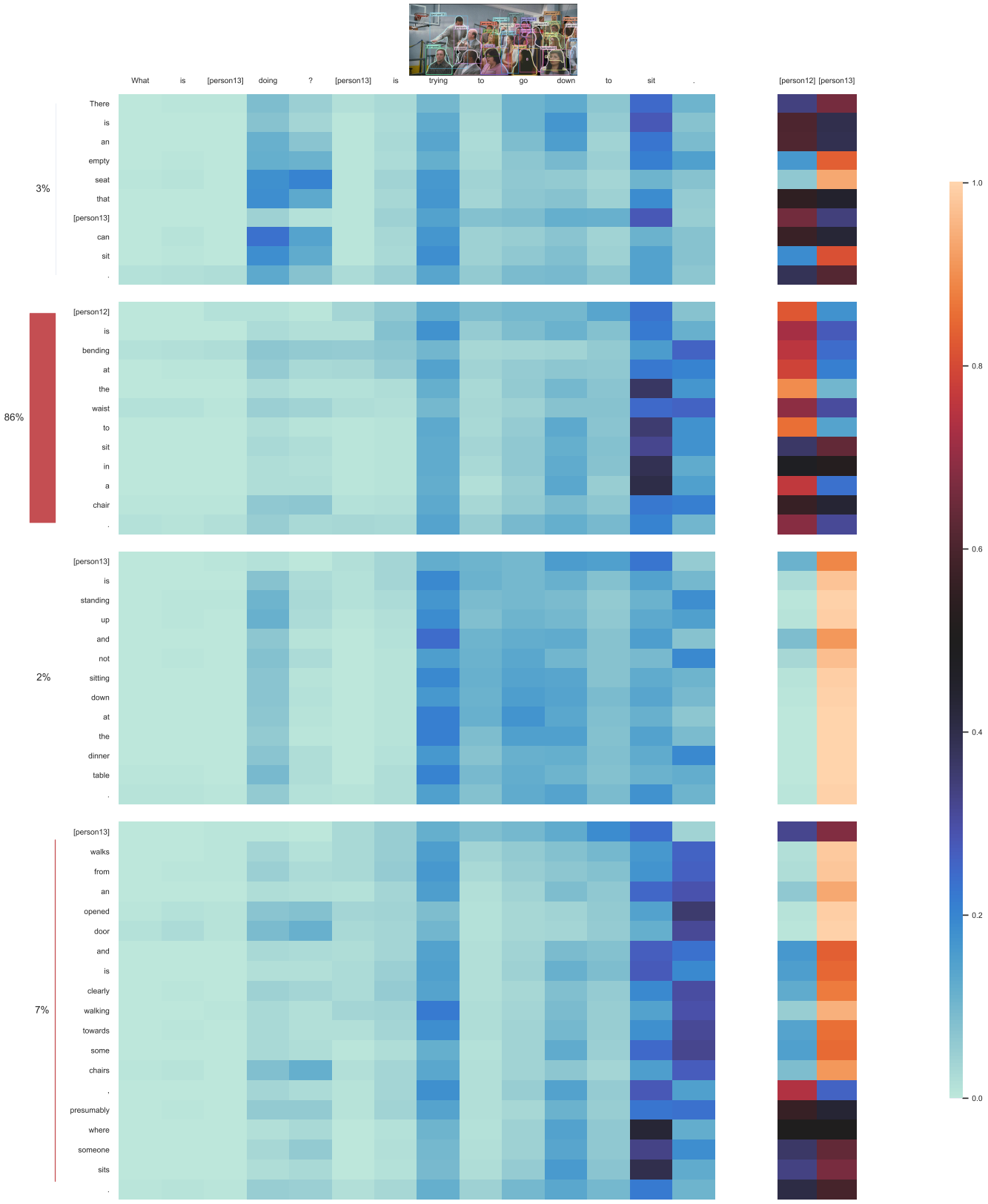


Figure 17: An example from the $QA \rightarrow R$ task. Each super-row is a response choice (four in total). The first super-column is the query, and the second super-column holds the relevant objects. Each block is a heatmap of the attention between each response choice and the query, as well as the attention between each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 86% confident that the right answer is b., which is incorrect - the correct answer is **a.**

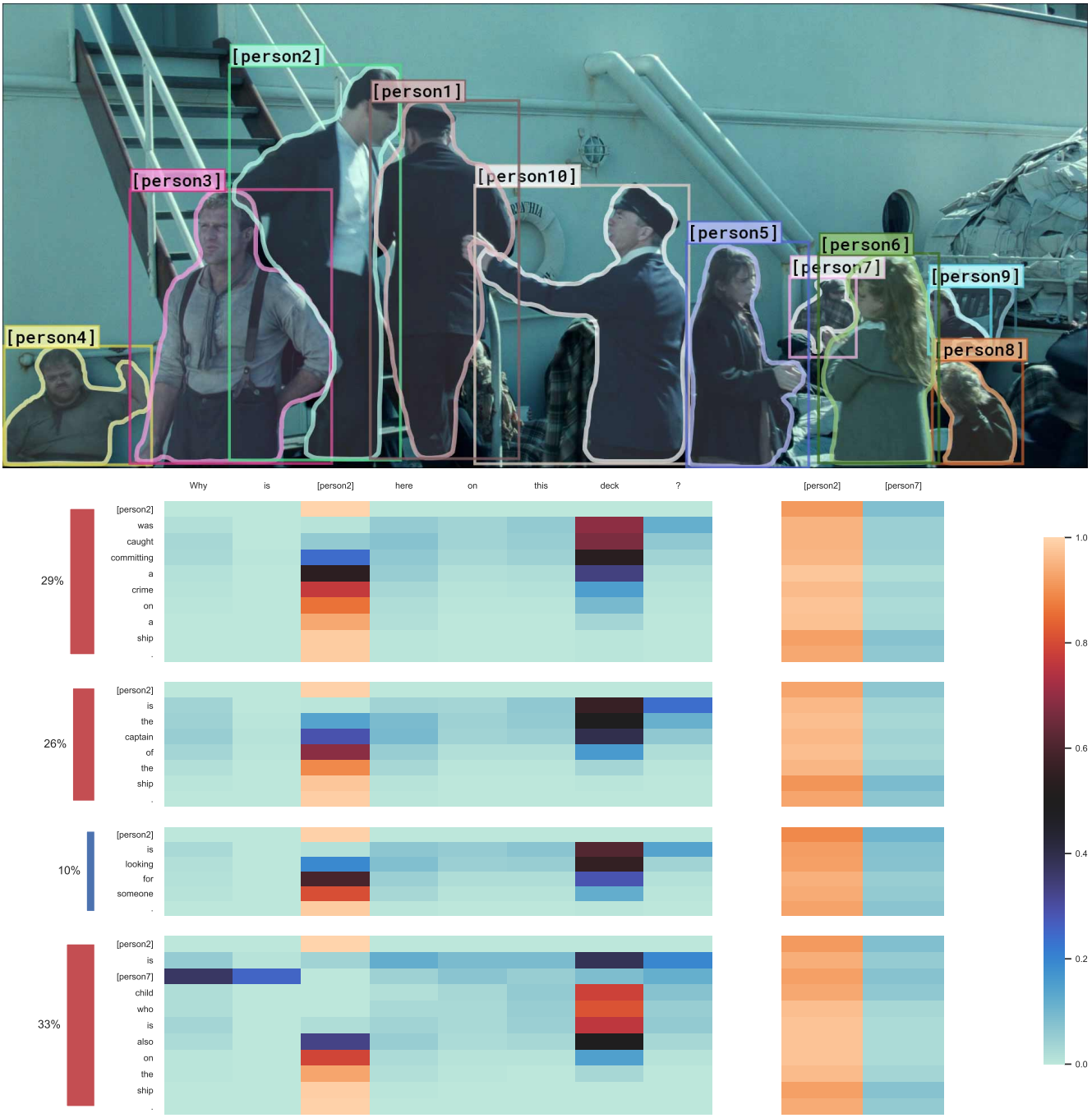


Figure 18: An example from the $Q \rightarrow A$ task. Each super-row is a response choice (four in total). The first super-column is the question: Here, 'Why is `[person2] here on this deck?`' and the second super-column represents the relevant objects in the image that **R2C** attends to. Accordingly, each block is a heatmap of the attention between each response choice and the query, as well as each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 33% confident that the right answer is d., which is incorrect - the correct answer is correct answer is **c.**

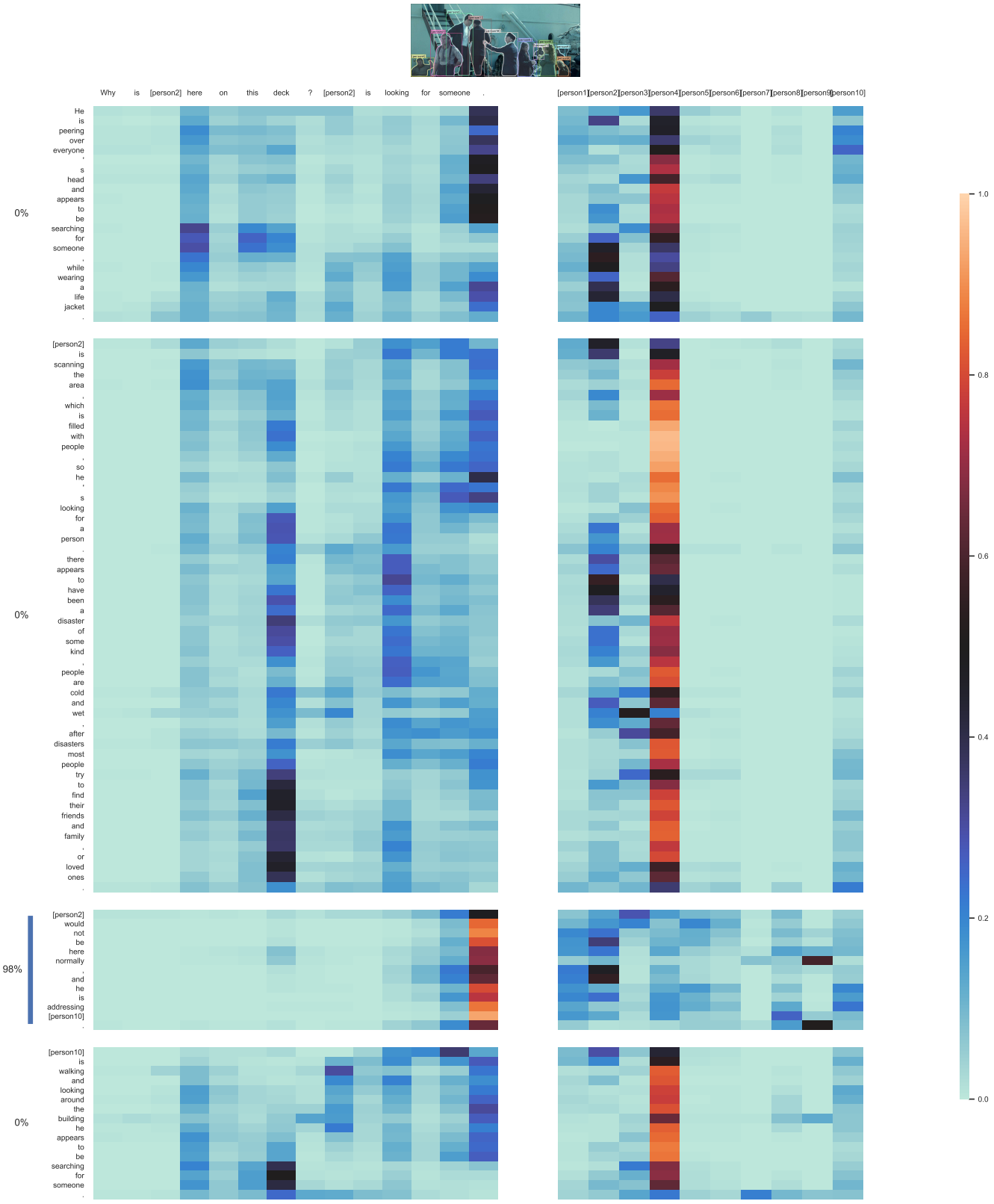


Figure 19: An example from the $QA \rightarrow R$ task. Each super-row is a response choice (four in total). The first super-column is the query, and the second super-column holds the relevant objects. Each block is a heatmap of the attention between each response choice and the query, as well as the attention between each response choice and the objects. The final prediction is given by the bar graph on the left: The model is 98% confident that the right answer is **c.**, which is correct.

References

[1] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Dont just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018. 4

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 3

[3] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 3

[4] 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. 6, 7

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3, 4, 7, 14

[6] H. Ben-younes, R. Cadene, M. Cord, and N. Thome. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6, 7

[7] O. Biran and C. Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, page 8, 2017. 3

[8] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642, 2015. 13, 15

[9] A. Chaganty, S. Mussmann, and P. Liang. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 643–653, 2018. 5

[10] A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh. Do explanations make vqa models more predictable to a human? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1036–1042, 2018. 3

[11] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668, 2017. 2, 5, 7, 13

[12] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014. 7

[13] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler. Learning to act properly: Predicting and explaining affordances from images. In *CVPR*, 2018. 3

[14] Y. Cui, G. Yang, A. Veit, X. Huang, and S. Belongie. Learning to evaluate image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5804–5812, 2018. 5

[15] E. Davis and G. Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58:92–103, 2015. 2

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 16

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 5, 6, 7, 13, 15, 16

[18] J. Devlin, S. Gupta, R. B. Girshick, M. Mitchell, and C. L. Zitnick. Exploring nearest neighbor approaches for image captioning. *CoRR*, abs/1505.04467, 2015. 4

[19] K. Ehsani, H. Bagherinezhad, J. Redmon, R. Mottaghi, and A. Farhadi. Who let the dogs out? modeling dog behavior from visual data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[20] P. Felsen, P. Agrawal, and J. Malik. What will happen next? forecasting player moves in sports videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3342–3351, 2017. 3

[21] A. Flowers. The Most Common Unisex Names In America: Is Yours One Of Them?, June 2015. 13, 16

[22] D. F. Fouhey, W. Kuo, A. A. Efros, and J. Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018. 4

[23] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016. 16

[24] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017. 14

[25] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 2, 4, 11

[26] 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*, volume 1, page 9, 2017. 4

[27] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 14

[28] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative

adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[29] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. Annotation artifacts in natural language inference data. In *Proc. of NAACL*, 2018. 2, 4, 5, 14, 15

[30] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 2, 4, 6, 11

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 11, 16

[32] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016. 3

[33] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata. Grounding visual explanations. *European Conference on Computer Vision (ECCV)*, 2018. 3

[34] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[35] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 5

[36] R. Hu, J. Andreas, T. Darrell, and K. Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. 3

[37] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4418–4427. IEEE, 2017. 3

[38] D. Huk Park, L. Anne Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[39] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer, 2016. 6, 7

[40] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Honolulu, Hawaii*, pages 2680–8, 2017. 15

[41] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987. 5

[42] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata. Textual explanations for self-driving vehicles. In *15th European Conference on Computer Vision*, pages 577–593. Springer, 2018. 3

[43] 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 *The 5th International Conference on Learning Representations*, 2017. 6, 7

[44] K. Kim, C. Nan, M. Heo, S. Choi, and B. Zhang. Pororoqa: Cartoon video series dataset for story understanding. In *Proceedings of NIPS 2016 Workshop on Large Scale Computer Vision System*, 2016. 15

[45] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 13, 17

[46] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 3

[47] J. Lei, L. Yu, M. Bansal, and T. L. Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018. 3, 4, 5, 15

[48] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650, 2016. 15

[49] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 17

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 5, 9, 10

[51] X. Lin and D. Parikh. Leveraging visual question answering for image-caption ranking. In *European Conference on Computer Vision*, pages 261–277. Springer, 2016. 7

[52] T. Maharaj, N. Ballas, A. Rohrbach, A. C. Courville, and C. J. Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[53] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 3

[54] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*, 2016. 14

[55] R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi. what happens if... learning to predict the effect of forces in images. In *European Conference on Computer Vision*, pages 269–285. Springer, 2016. 3

[56] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 5

[57] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 13

[58] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237, 2018. 2, 5, 7, 13

[59] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the why in images. *arXiv preprint arXiv:1406.5472*, 2014. 3

[60] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proc. ICCV*, 2017. 3

[61] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 3

[62] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme. Hypothesis Only Baselines in Natural Language Inference. *arXiv:1805.01042 [cs]*, May 2018. arXiv: 1805.01042. 2, 5

[63] S. Ramakrishnan, A. Agrawal, and S. Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, 2018. 4

[64] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6

[65] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017. 3

[66] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016. 3

[67] A. Rohrbach, M. Rohrbach, S. Tang, S. J. Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE. 3

[68] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie Description. *International Journal of Computer Vision*, 123(1):94–120, May 2017. 4, 9, 11, 12

[69] R. Rudinger, C. May, and B. Van Durme. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, 2017. 14

[70] M. Sap, M. C. Prasettio, A. Holtzman, H. Rashkin, and Y. Choi. Connotation frames of power and agency in modern films. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2329–2334, 2017. 14

[71] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 16

[72] A. Schofield and L. Mehr. Gender-distinguishing features in film dialogue. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 32–39, 2016. 14

[73] R. Schwartz, M. Sap, I. Konstas, L. Zilles, Y. Choi, and N. A. Smith. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proc. of CoNLL*, 2017. 2, 5, 14

[74] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics. 13

[75] K. K. Singh, K. Fatahalian, and A. A. Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 3

[76] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. 3, 4, 5, 15

[77] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011. 4, 14

[78] P. Vicol, M. Tapaswi, L. Castrejon, and S. Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[79] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016. 3

[80] M. Wagner, H. Basevi, R. Shetty, W. Li, M. Malinowski, M. Fritz, and A. Leonardis. Answering visual what-if questions: From actions to predicted scene descriptions. In *Visual Learning and Embodied Agents in Simulation Environments Workshop at European Conference on Computer Vision*, 2018. 3

[81] P. Wang, Q. Wu, C. Shen, A. van den Hengel, and A. R. Dick. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 3

[82] J. Wieting, J. Mallinson, and K. Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, 2017. 13

[83] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long*

Papers), pages 1112–1122. Association for Computational Linguistics, 2018. 13

[84] Q. Wu, P. Wang, C. Shen, A. R. Dick, and A. van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4622–4630, 2016. 3

[85] T. Ye, X. Wang, J. Davidson, and A. Gupta. Interpretable intuitive physics model. In *European Conference on Computer Vision*, pages 89–105. Springer, 2018. 3

[86] Y. Yoshikawa, J. Lin, and A. Takeuchi. Stair actions: A video dataset of everyday home actions. *arXiv preprint arXiv:1804.04326*, 2018. 3

[87] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual Madlibs: Fill in the blank Image Generation and Question Answering. *arXiv:1506.00278 [cs]*, May 2015. arXiv: 1506.00278. 3

[88] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 3

[89] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speakerlistener-reinforcer model for referring expressions. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[90] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 4

[91] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, 2017. 14

[92] L. Zhou, C. Xu, and J. J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 3

[93] Y. Zhou and T. L. Berg. Temporal perception and prediction in ego-centric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4498–4506, 2015. 3

[94] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7W: Grounded Question Answering in Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4

[95] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015. 16