7. Supplementary

The supplementary sections provide more detail on the methods and experiments described in our paper. First, we explain in more detail our process for decomposing questions into hierarchies, including two human studies. We provide equations for our metrics, then describe our training process for the experiments, explore how AGQA-Decomp performs as data augmentation for AGQA, add results for the Most Likely baseline, and present example error modes we found through qualitative analysis of hierarchies. Finally, we discuss directions for future work.

7.1. Dataset

In this section, we provide additional details for the process of question decomposition. We first describe the process of generating individual sub-question hierarchies from AGQA programs. We then explain how we obtain answers for these questions, first detailing the general case and then describing the edge case of Object Exists questions. We finally discuss the limitations and potential societal impact of our hierarchies and report human performance measured through AMT studies.

AGQA version. We use an updated version of AGQA[1] that incorporates multiple improvements over the original dataset [5]. Throughout the paper we refer to this updated version as AGQA for simplicity. The most significant change is an updated balancing algorithm to further reduce linguistic biases. Some smaller improvements were motivated by minor errors in AGQA we discovered while ensure that AGQA-Decomp was internally consistent.

AGQA program to subquestion hierarchy. In order to generate sub-question hierarchies, we first convert the original AGQA programs to a new program format. Each AGQA question type has a simple program template associated with it. To get compositional questions, AGQA makes this template more complex by introducing indirect references and temporal localization. As such, while forming the new programs, we firstly get the smaller programs for indirect references, if there are any, and continue by getting the temporal localization and the simple program associated with the basic template. We finally combine these to form the new program.

For example, the AGQA type focusing on the existence of a relation between a person and an object has `Exists([object], Iterate(video, Filter(frame, [relations, [relation], objects])))` as its simple program. Using this structure of the AGQA original program template, we extract the object and relation for the new program. In this simple form, the corresponding new program is `interactionExists(objExists(person), relationExists([relation]), objExists([object]))`. We perform a similar process of translating between program types for temporal localization phrases (e.g. `Localize(before, action)` translates into `before(..., action program)`). Step 1 of Figure 5 visualizes the conversion of an AGQA program to the new program format.

Upon converting AGQA programs into the new program format, we derive subquestion hierarchies from the new programs. Step 2 of Figure 5 and Algorithm 1 illustrate the decomposition process for the newly generated program.

Use of the unbalanced AGQA dataset. Given question decompositions, our first strategy for obtaining ground-truth answers is to rely on the original AGQA annotations. This approach is not straightforward. After decomposing the questions in the balanced AGQA dataset, we can find sub-questions that are not present in the balanced dataset.

Algorithm 1: Question hierarchy generation

```
Input: $p$: Question program
Output: Question decomposition hierarchy
def main($p$):
    $V$ = empty set for vertices
    $E$ = empty set for edges
    buildDAG($p$)

def buildDAG($p$):
    $subprograms$ = inner functions of $p$
    if no subprograms then
        $s$ = $p$'s natural language question equivalent
        Add $s$ to $V$
        $indirect$ = program phrase replacing $p$
        return $s$, $indirect$
    end

    $S_q$ = empty set for subquestions
    for subprogram in subprograms do
        $s$, $indirect$ = buildDAG($subprogram$)
        Add $s$ to $S_q$
        $p$ = $p$ replacing $subprogram$ with $indirect$
    end

    $q$ = $p$'s natural language question
    Add $q$ to $V$
    for $s$ in $S_q$ do
        Add ($q$, $s$, composition) to $E$
    end
    return question, reference
```

[1] AGQA 2.0: https://tinyurl.com/agqavideo

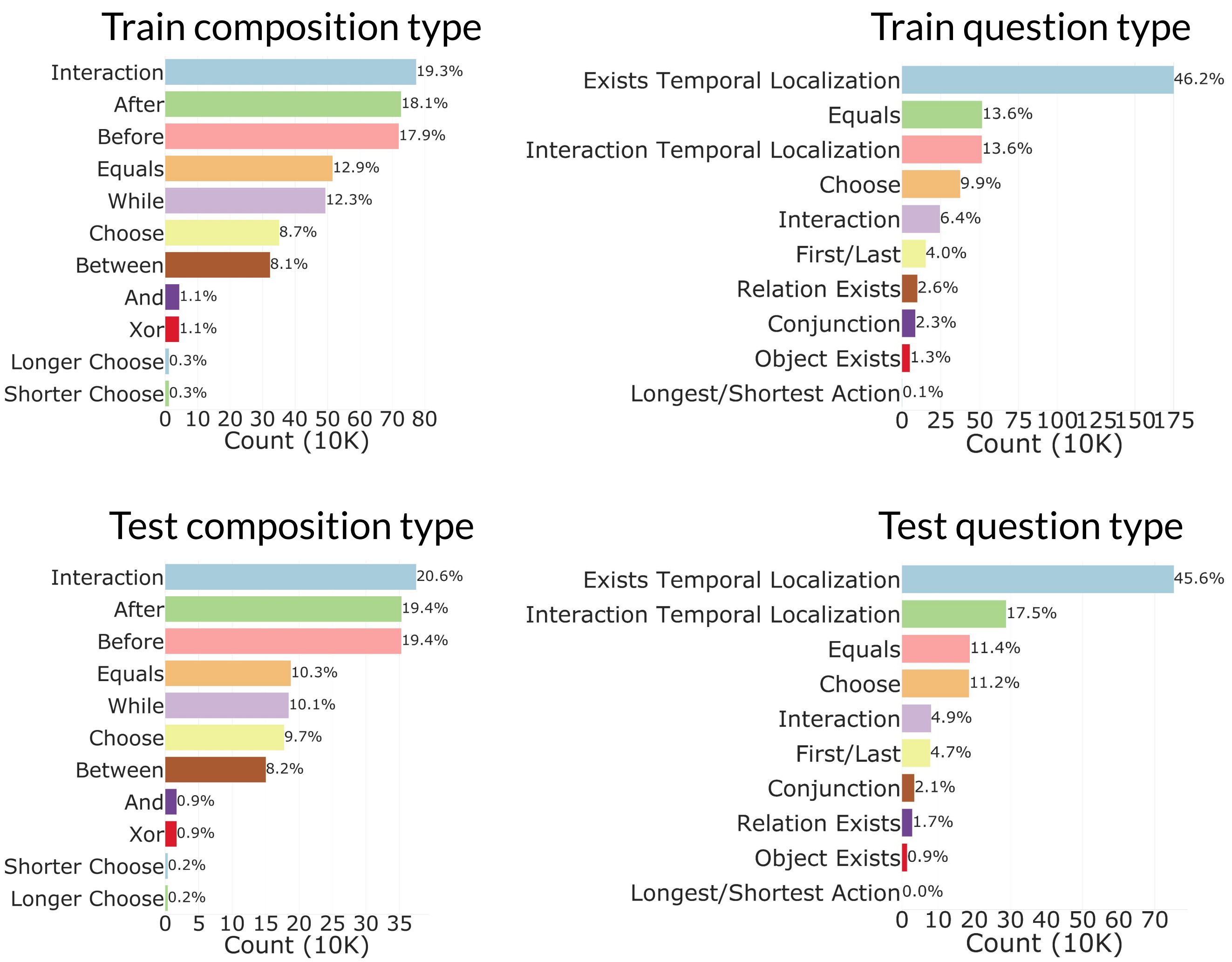


Figure 4. **Left:** A bar chart displaying the distribution of composition rule types on the test set. Interaction and After composition rules are the most common. **Right:** A bar chart displaying the distribution of question types on the test set. Exists temporal localization questions dominate the test set.

If the parent is present in the balanced AGQA dataset, we are not guaranteed that the sub-question will also be present in the balanced version and cannot use the balanced dataset to derive its answer. Since the unbalanced AGQA dataset covers most of the possible questions (other than the newly added exists sub-questions), we rely on it to get answers for the question decompositions in the balanced version.

Specifically, we decompose 97M questions from the unbalanced AGQA dataset, with each hierarchy having an average of 16.81 sub-questions. This process produces 25.53M unique new sub-questions. We determine the answers using different logical consistency rules for each video in the unbalanced dataset, which answers 87.92% of the data. The question-answer pairs for a video from the decomposed unbalanced AGQA dataset are then used to answer our questions and our sub-questions for the same video. This process generates answers for in 90.23% of the data in our balanced subquestion hierarchies.

No exists questions. Object Exists questions (e.g. "Does a closet exist?") are not a part of the original AGQA dataset. This is because the Action Genome scene graphs, which were used to generate the AGQA questions, only contain objects that an actor is interacting with [9]. Therefore, existing objects that are in the background, or that are extremely common (e.g. clothes, floor), are often not annotated. We can infer the "yes" answer through logical entailment (e.g. if the answer to the question "Did they interact with <object>?" is "yes", then its sub-question "Does <object> exist?" must also be "yes"). However, there is no way to use logical entailments to determine which objects

Table 5. We handcraft logical consistency rules that check whether a model is consistent when answering questions in a DAG. The rules are implications, i.e if q has answer a then $s1$ should be b.

Composition	**Consistency Rules**	**Example**
`Interaction`	If an interaction '[person] [relation] [object]' is 'Yes' its direct sub-questions '[person]' exist, '[relationship]' exist and [object]' should be 'Yes'	*q* : Is a person holding a dish? – Yes *s1*: Does a person exist? – Yes *s2*: Is a person holding something? – Yes *s3*: Does a dish exist? – Yes
`Temporal localization`	If '[exists question] [temporal localization] [condition]' is 'Yes' then '[exists question]' is 'Yes' and '[condition]' is 'Yes'	*q* : Does a person exist after smiling at something? – Yes *s1*: Does a person exist? – Yes *s2*: Is a person smiling at something? – Yes
`And`	If '[action1] and [action2]' is 'Yes' then '[action1]' should be 'Yes' and '[action2]' should be 'Yes'	*q* : Is the person holding a cup and touching a dish? – Yes *s1*: Is the person holding a cup? – Yes AND *s2*: Is the person touching a dish? – Yes
	If '[action1] and [action2]' is 'No' then either [action1] should be 'No' or [action2] should be 'No'	*q* : Is the person touching a bottle and opening a window? – No *s1*: Is the person touching a bottle? – No **OR** *s2*: Is the person opening a window? – No
`Xor`	If '[action1] but not [action2]' is 'Yes' then '[action1]' should be 'Yes' and '[action2]' should be 'No'	*q* : Is the person smiling at something but not walking through a doorway? – Yes *s1*: Is the person smiling at something? – Yes *s2*: Is the person walking through a doorway? – No
	If '[action1] but not [action2]' is 'No' then either [action1] should be 'No' or [action2] should be 'Yes'	*q* :Is the person throwing a cup but not leaning on the doorway? – No *s1*: Is the person throwing a cup? – No **OR** *s2*: Is the person leaning on a doorway? – Yes
`Equals`	If '[object] equals [indirect object]' is 'Yes' then '[indirect object]' should be '[object]' and '[object]' exists is 'Yes'	*q* : Is a doorway the first object they are holding? – Yes *s1*: Which is the first object they are holding? – doorway *s2*: Does a doorway exist? – Yes
	If '[object] equals [indirect object]' is 'No' then [indirect object] should not be [object]	*q* : Is the book the last object that they are putting? – No *s1*: Which is the last object that the person is putting? – **NOT** book
`Choose (Objects/ time)`	If 'choose [object1] or [object2] [indirect object]' is 'object1' then [object1] equals [indirect object] should be 'Yes' and [object2] equals [indirect object] should be 'No'	*q* : Is the doorway or the cup the first object they went behind? – doorway *s1*: Is the doorway the first object they went behind? – Yes *s2*: Is the cup the first object they went behind? – No
	If 'Does [action1] occur before or after [action2]' is 'before' then 'Does [action1] occur before [action2]?' should be 'Yes' and 'Does [action1] occur after [action2]?' should be 'No'	*q* : Is the person holding a cup before or after smiling at something? – before *s1*: Is the person holding a cup before smiling at something? – Yes *s2*: Is the person holding a cup after smiling at something? – No

do not exist.

Therefore, we generate Object Exists questions answered "no" through two methods. First, we source human annotations for what objects do not exist within the video (see Human evaluation subsection). Then, we also include questions in which the object exists, but the temporal localization phrase contains an invalid action ("Does <object> exist before they <invalid action>?"). These two methods generate $135K$ Object Exists questions with a "no" answer.

Limitations. There are limitations to our approach. First, this approach assumes AGQA answers to be ground truth. However, like all benchmarks, AGQA answers can be incorrect. These errors are described in more detail in their paper [5].

Furthermore, not all questions in the hierarchies can be answered by the scene graph annotations AGQA uses as its basis for video representation [9, 11]. The AGQA scene graphs only annotate objects with which the actor is interacting, so they may miss existing objects in the background of the video or objects that are so generic that they often exist without annotations (e.g. "floor" or "clothes"). The blacklisting of certain questions in AGQA also affected the subset of sub-questions in our decompositions that have associated AGQA answers.

Societal impact. Large curated datasets used to train vision models are known to contain biases, be it gender [7, 17], racial [2, 16] or geographic [12], or with problematic content [1]. Models trained on these datasets can then learn and propagate these biases to the real world, causing unintended harm. We note that AGQA-Decomp is primarily intended as a diagnostic dataset guiding model development and evaluation. A user leveraging AGQA-Decomp as training data should therefore recognize that models can propagate biases latent in the training data. Furthermore, as detailed in the limitation section, our automatic generalization process propagates error for ground-truth answers in the AGQA dataset, which can hurt real-world performance.

Collecting more annotations. To identify missing objects from the scene graphs, we create an object labeling task. When we know that an object definitively doesn't exist, we can now answer questions that have the answer "no" (e.g. "Did the person touch a cup?" would be "no" if cup was not identified anywhere in the video.). We pay such that the equivalent hourly rate is $15 per hour.

The question decomposition method cannot infer Object Exists questions with the answer "no." Therefore, we run a study for human participants to mark which objects do not exist in the video. For a given video, participants must select the objects that do not appear in the video from a list of nearly all the objects in AGQA (See Figure 8). We do not offer objects that nearly always exist (person, clothes, floor, hands, and hair). We quality check by looking at whether they mark objects in the scene graph as present in the video.

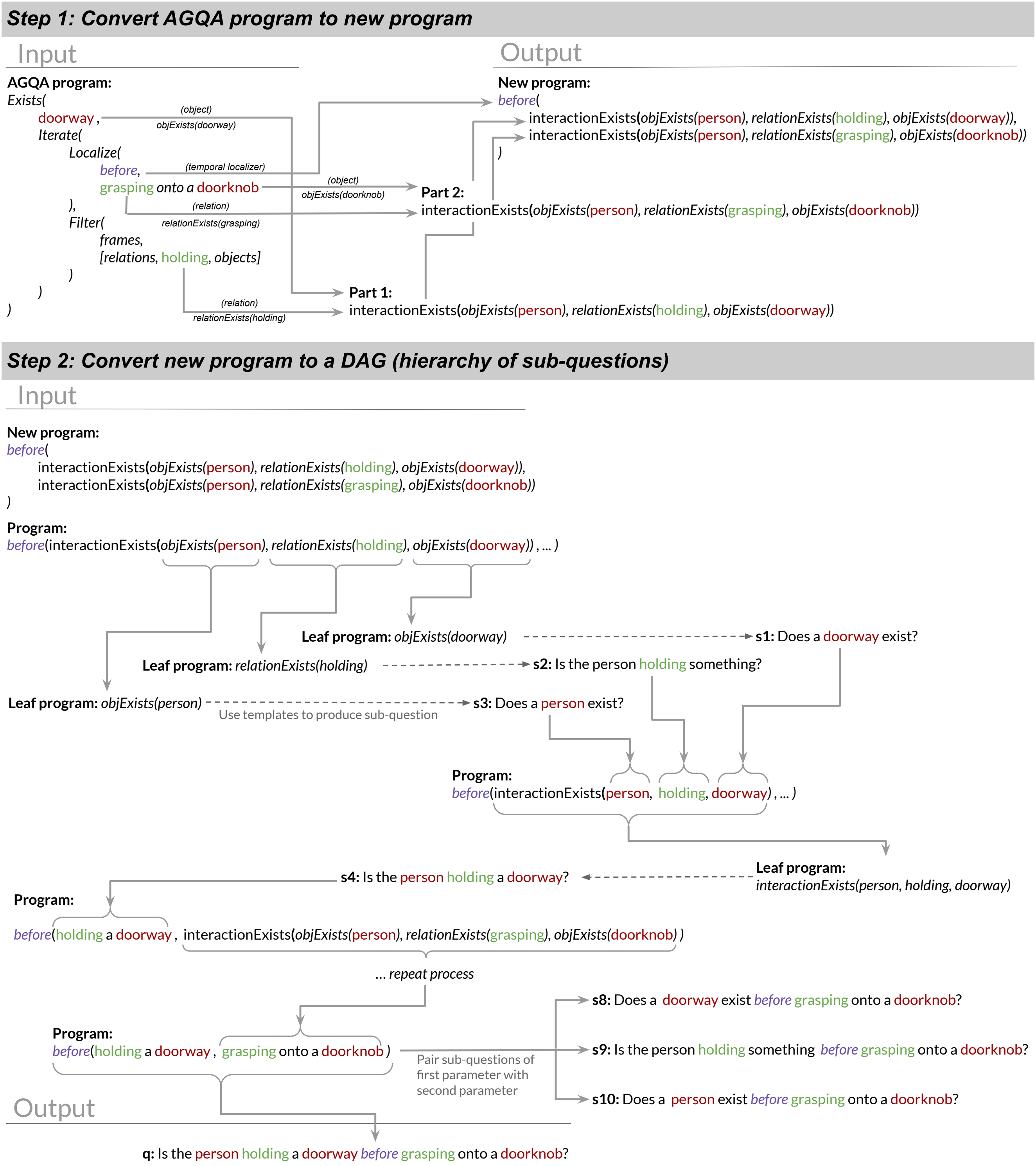
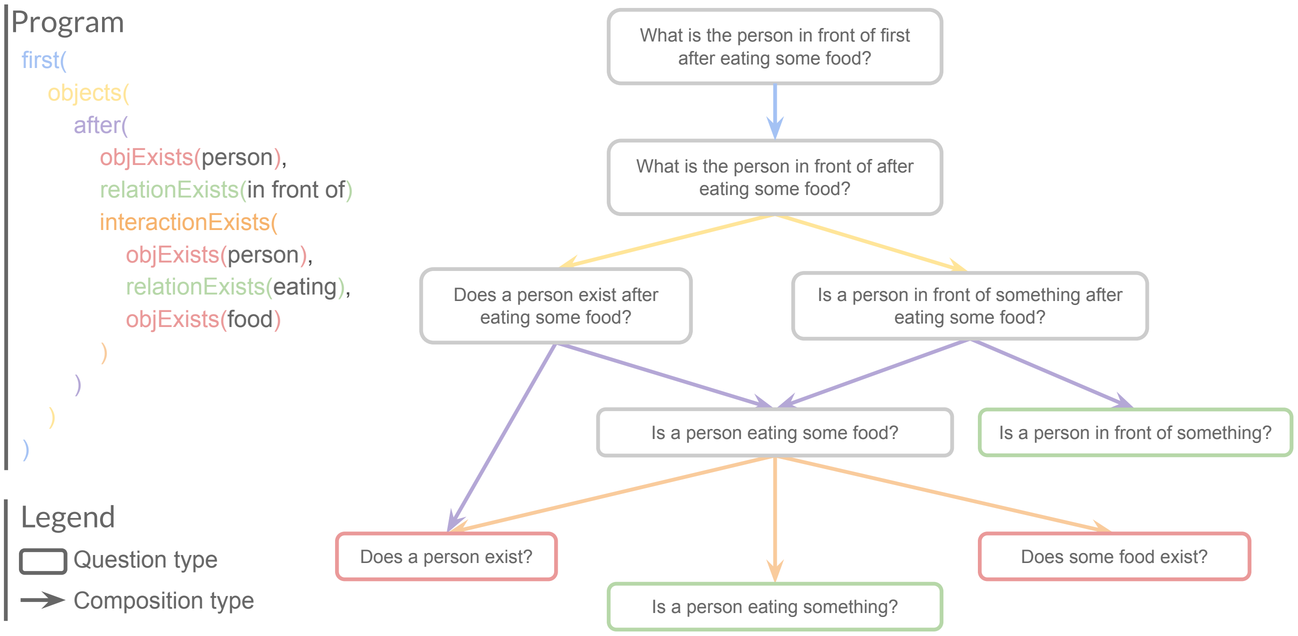
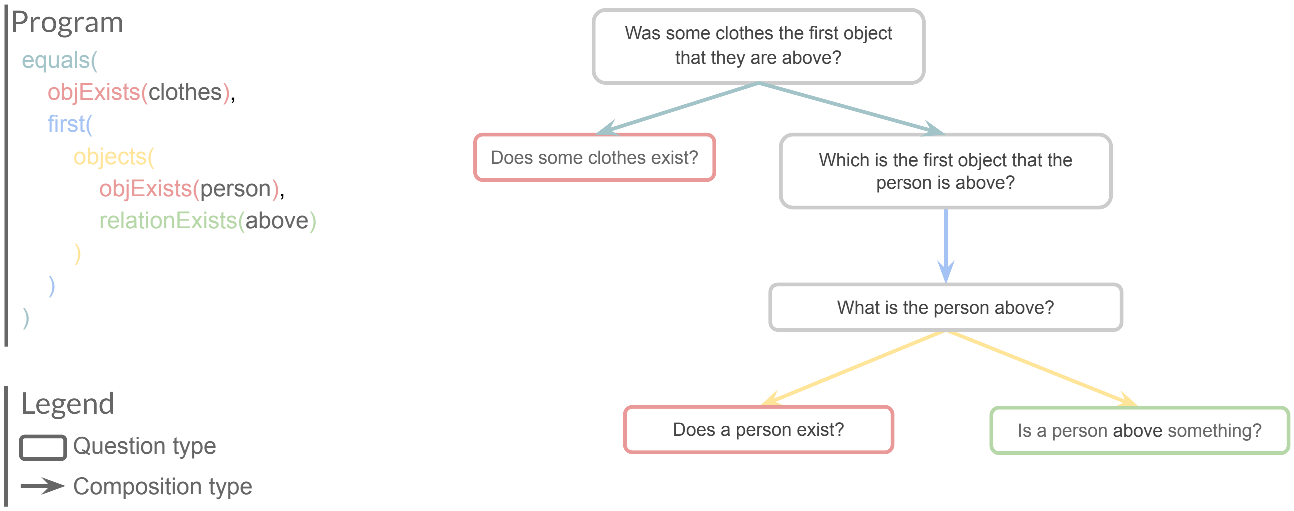

Step 1: Convert AGQA program to new program
Input
AGQA program:
Exists(
doorway ,
Iterate(
Localize(
before,
grasping onto a doorknob
),
Filter(
frames,
[relations, holding, objects]
)
)
)
(object)
objExists(doorway)
(temporal localizer)
(object)
objExists(doorknob)
(relation)
relationExists(grasping)
(relation)
relationExists(holding)
Output
New program:
before(
interactionExists(objExists(person), relationExists(holding), objExists(doorway)),
interactionExists(objExists(person), relationExists(grasping), objExists(doorknob))
)
Part 2:
interactionExists(objExists(person), relationExists(grasping), objExists(doorknob))
Part 1:
interactionExists(objExists(person), relationExists(holding), objExists(doorway))
Step 2: Convert new program to a DAG (hierarchy of sub-questions)
Input
New program:
before(
interactionExists(objExists(person), relationExists(holding), objExists(doorway)),
interactionExists(objExists(person), relationExists(grasping), objExists(doorknob))
)
Program:
before(interactionExists(objExists(person), relationExists(holding), objExists(doorway)) , ...)
Leaf program: objExists(doorway)
s1: Does a doorway exist?
Leaf program: relationExists(holding)
s2: Is the person holding something?
Leaf program: objExists(person)
s3: Does a person exist?
Use templates to produce sub-question
Program:
before(interactionExists(person, holding, doorway) , ...)
Leaf program:
interactionExists(person, holding, doorway)
s4: Is the person holding a doorway?
Program:
before(holding a doorway , interactionExists(objExists(person), relationExists(grasping), objExists(doorknob))
... repeat process
Program:
before(holding a doorway , grasping onto a doorknob)
Pair sub-questions of first parameter with second parameter
s8: Does a doorway exist before grasping onto a doorknob?
s9: Is the person holding something before grasping onto a doorknob?
s10: Does a person exist before grasping onto a doorknob?
Output
q: Is the person holding a doorway before grasping onto a doorknob?

Figure 5. The figure shows the process of generating a question hierarchy using an AGQA program for the example AGQA question "Is the person holding a doorway before grasping onto a doorknob?" **Step 1:** We transform the AGQA program into a program representing the reasoning steps of the question. **Step 2:** We use Algorithm 1 to generate the hierarchy of sub-questions from the new program.


Program
Legend
Question type
Composition type
What is the person in front of first after eating some food?
What is the person in front of after eating some food?
Does a person exist after eating some food?
Is a person in front of something after eating some food?
Is a person eating some food?
Is a person in front of something?
Does a person exist?
Is a person eating something?
Does some food exist?

```
first(
  objects(
    after(
      objExists(person),
      relationExists(in front of)
      interactionExists(
        objExists(person),
        relationExists(eating),
        objExists(food)
      )
    )
  )
)
```

Figure 6. Example decomposition with corresponding program.


Program
Legend
Question type
Composition type
Was some clothes the first object that they are above?
Does some clothes exist?
Which is the first object that the person is above?
What is the person above?
Does a person exist?
Is a person above something?

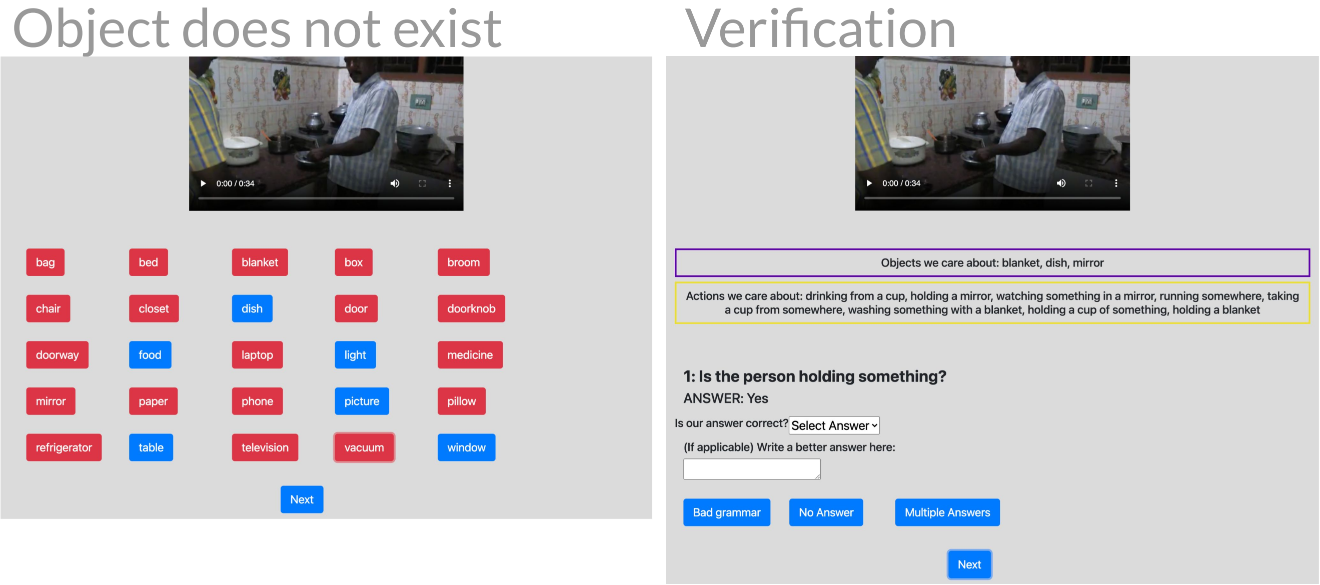
```
equals(
  objExists(clothes),
  first(
    objects(
      objExists(person),
      relationExists(above)
    )
  )
)
```

Figure 7. Example decomposition with corresponding program.

At the end of this process, we have the objects that do not exist for 88 randomly selected videos. We were not able to to repeat this process on all videos due to monetary and time restrictions. We then take these objects and use the subquestion templates to generate questions. We use actions within the video to also generate questions with temporal localization phrases. This process generates $135K$ Object Exists questions with a "No" answer.

Human evaluation. We evaluate the accuracy of sub-questions to find the error rate in each sub-question type.

Our answers in the question decompositions originate from the AGQA dataset as well as from logical entailments. Therefore, the errors that our human annotators mark in the questions originate from the AGQA dataset. The AGQA benchmark paper provides details about the source of these errors, including incorrect annotations, incorrect augmentations, inconsistent annotations, and human-AGQA definition mismatches [5]. We run the same validation task as the



Objects we care about: blanket, dish, mirror

Actions we care about: drinking from a cup, holding a mirror, watching something in a mirror, running somewhere, taking a cup from somewhere, washing something with a blanket, holding a cup of something, holding a blanket

1: Is the person holding something?

ANSWER: Yes

Is our answer correct? Select Answer

(If applicable) Write a better answer here:

Figure 8. **Left:** The annotator views a video and a list of nearly all the objects in AMT. Annotators select the objects that do not appear in the video. **Right:** The annotator watches five videos that each appear with a question and an answer. Annotators indicate if that answer is Correct or Incorrect from a dropdown menu.

AGQA benchmark on at least 25 questions per sub-question type. For all analysis, we take the the majority vote of 3 annotators for each question.

In this task annotators see a question, answer, and video. They are provided with a dropdown menu to mark the question as Correct or Incorrect. If they select Incorrect, we provide a space to write the correct answer. We also collect information on whether the question has bad grammar, multiple answers, or no possible answers. We check for the quality of responses with questions that we know to be answered incorrectly. Annotators mark 88.00% of these incorrect questions as incorrect.

7.2. Metrics

In this section, we give additional details for our metrics. We first provide precise definitions for each metric. Afterwards, we give guidelines on how to interpret and compare values for each metric.

In Section 4 of the main paper, we gave definitions for our metrics in plain English. We provide equations for each for further clarity. In all following definitions, let f refer to the model we want to evaluate. Given an input video-question pair (v, q), we set $\text{Acc}(v, q, f) = 1$ if f made a correct prediction on this input and 0 otherwise.

Compositional accuracy (CA): We will begin with a formal definition of the metric's general form. Let q be an arbitrary question and define C_q to be the set of immediate sub-questions associated with q. To compute CA, we consider the set Q_{CA} of all video-question pairs (v, q) where $|C_q| > 0$ and $\text{Acc}(v, s, f) = 1$ for all $s \in C_q$. Then,

$$\text{CA}(f) = \frac{\sum_{(v,q)\in Q_{CA}} \text{Acc}(v, q, f)}{|Q_{CA}|}.$$

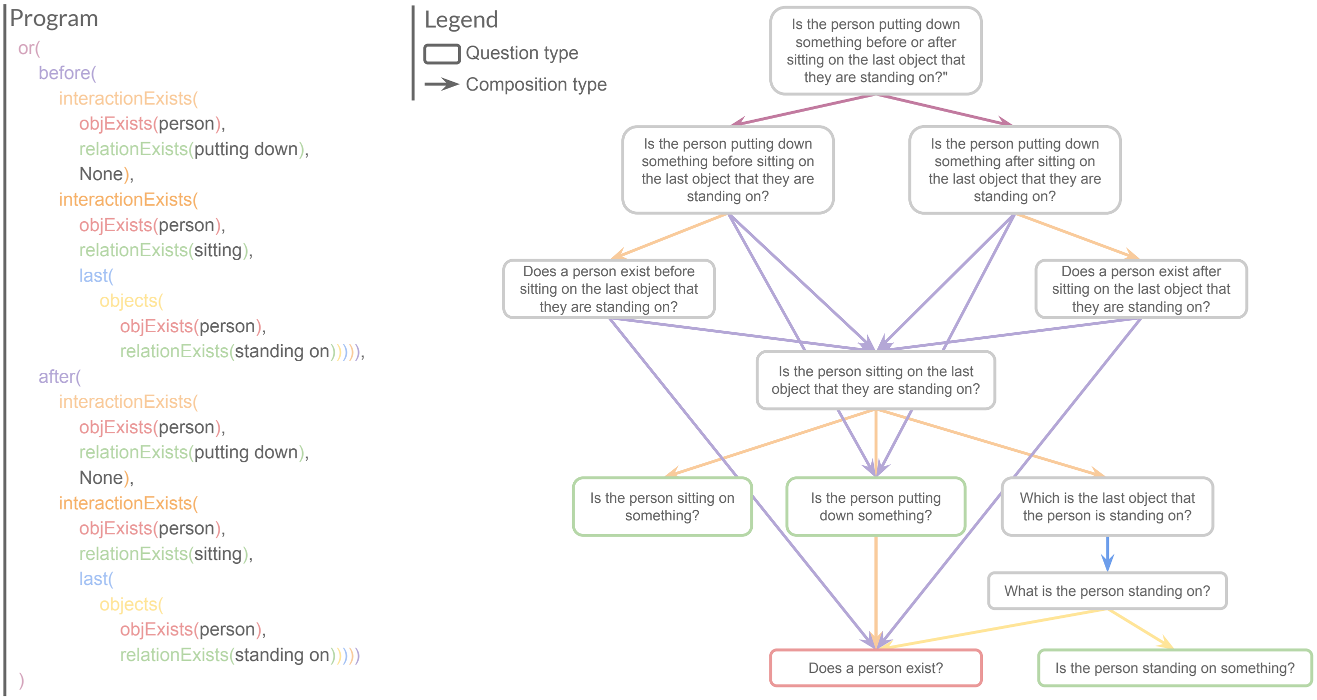
When we condition on question types, we compute the average on a subset of Q_{CA} where the parent questions q belong to a particular question type p instead. The change is more complicated when we condition on composition rules, however. Let t be the composition rule we are conditioning on. Then, for each question q, we change all instances of C_q to $C_{q,t} = \{s \in C_q | (q, s, t) \in E_q\}$, where E_q is the set of edges in the DAG associated with q. In plain English, we consider only the immediate sub-questions of q related to it by the composition rule t.

Right for the wrong reasons (RWR): The formulas for RWR are similar to those for CA. To compute RWR, we consider the set Q_{RWR} of all video-question pairs (v, q) where $|C_q| > 0$ and where there exists at least one $s \in C_q$ such that $\text{Acc}(v, s, f) = 0$. Then,

$$\text{RWR}(f) = \frac{\sum_{(v,q)\in Q_{RWR}} \text{Acc}(v, q, f)}{|Q_{RWR}|}.$$

We condition on question types and on composition rules using the exact method as for CA.

To compute the more granular variant of RWR, RWR-n, we perform the same operations on the set Q_{RWR-n} of all



```
Program
or(
    before(
        interactionExists(
            objExists(person),
            relationExists(putting down),
            None),
        interactionExists(
            objExists(person),
            relationExists(sitting),
            last(
                objects(
                    objExists(person),
                    relationExists(standing on))))),
    after(
        interactionExists(
            objExists(person),
            relationExists(putting down),
            None),
        interactionExists(
            objExists(person),
            relationExists(sitting),
            last(
                objects(
                    objExists(person),
                    relationExists(standing on)))))
)
```

Legend
Question type
Composition type
Is the person putting down something before or after sitting on the last object that they are standing on?"
Is the person putting down something before sitting on the last object that they are standing on?
Is the person putting down something after sitting on the last object that they are standing on?
Does a person exist before sitting on the last object that they are standing on?
Does a person exist after sitting on the last object that they are standing on?
Is the person sitting on the last object that they are standing on?
Is the person sitting on something?
Is the person putting down something?
Which is the last object that the person is standing on?
What is the person standing on?
Does a person exist?
Is the person standing on something?

Figure 9. Example decomposition with corresponding program.

video-questions pairs (v, q) where $|C_q| > 0$ and where the number of $s \in C_q$ such that $\text{Acc}(v, s, f) = 0$ is exactly n.

Delta: Delta is defined as the difference between RWR and CA for a given model f:

$$\text{Delta}(f) = \text{RWR}(f) - \text{CA}(f).$$

Internal Consistency (IC): We will begin with a formal definition of the metric's general form. Denote Φ as the set of all logical consistency rules. Let $\phi \in \Phi$ be any logical consistency rule, (v, q) be any arbitrary video-question pair and C_q be the set of immediate sub-questions associated with q. We then set $\phi(q, C_q, v, f) = 1$ if f's predictions for q and its sub-questions pass ϕ's consistency check, 0 if it fails and -1 if the check cannot be applied to the given set of question-answer pairs. In order to compute internal consistency for a given logical consistency rule $\phi \in \Phi$, denoted IC_ϕ, we consider the set Q^ϕ_{IC} of all video-question pairs (v, q) such that $\phi(q, C_q, v, f) \neq -1$. We then define

$$IC_\phi(f) = \frac{\displaystyle\sum_{(v,q)\in Q^\phi_{IC}} \phi(c, C_q, v, f)}{|Q^\phi_{IC}|}.$$

The overall IC metric is then defined as

$$IC(f) = \frac{\sum_{\phi\in\Phi} IC_\phi(f)}{|\Phi|}.$$

If any $IC_\phi(f)$ is undefined due to $|Q^\phi_{IC}| = 0$, we also treat $IC(f)$ as undefined.

In order to condition on a particular composition rule t for IC, we simply perform the same operations using the set of logical consistency rules Φ_t applicable to t instead of the general set Φ. Conditioning on a specific parent question type p is similar, but more complicated. As before, we restrict our attention to the set of logical consistency rules Φ_p applicable to the parent question type p. However, we further focus on subsets of Q^ϕ_{IC} where the parent questions q belong to the question type p.

Accuracy: We compute accuracy per question type and normalize across answers to obtain an aggregate value. Consider any question type t and let A_t be the set of ground-truth answers associated with questions of type t. Referring to $Q_{t,a}$ as the set of video-question pairs (v, q) where q is of type t and for which a is the ground-truth answer, we formally define

$$\text{Accuracy}(f, t) = \frac{\displaystyle\sum_{a\in A_t} \frac{\sum_{(v,q)\in Q_{t,a}} \text{Acc}(v, q, f)}{|Q_{t,a}|}}{|A_t|}.$$

Interpreting Values for Metrics. We expect a model that reasons compositionally to have high values for the Accuracy, CA, and IC metrics and to have low values for the

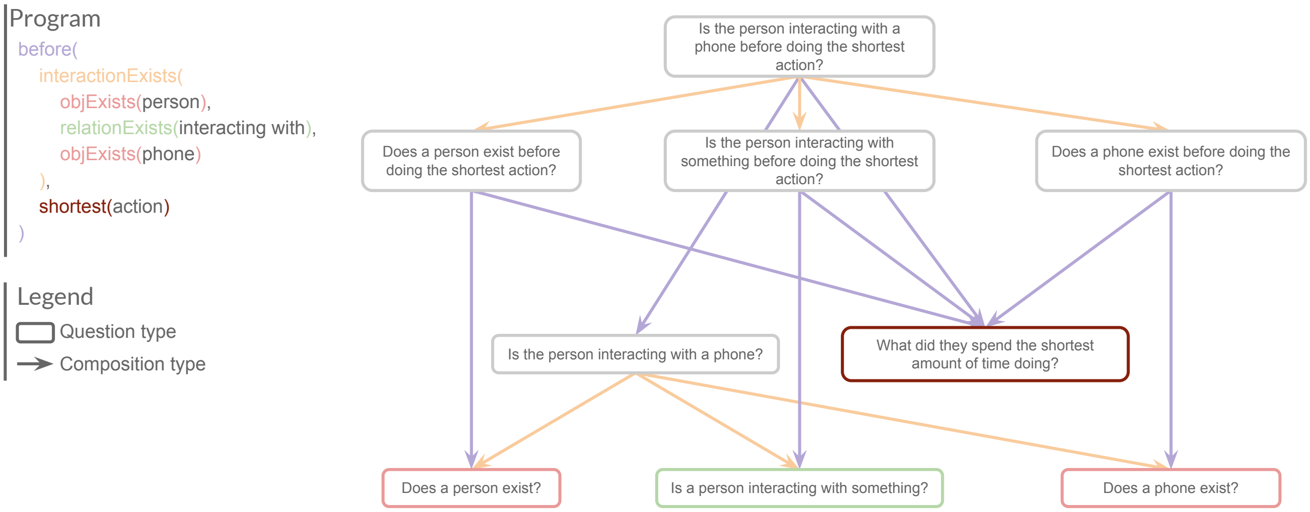

Program
before(
interactionExists(
objExists(person),
relationExists(interacting with),
objExists(phone)
),
shortest(action)
)
Legend
Question type
Composition type
Is the person interacting with a phone before doing the shortest action?
Does a person exist before doing the shortest action?
Is the person interacting with something before doing the shortest action?
Does a phone exist before doing the shortest action?
Is the person interacting with a phone?
What did they spend the shortest amount of time doing?
Does a person exist?
Is a person interacting with something?
Does a phone exist?

Figure 10. Example decomposition with its corresponding program.

RWR metric. Given that we expect a model to perform poorer on parent questions when it answers at least one subquestion incorrectly, we also expect a model that reasons compositionally to obtain negative Delta values. In other words, we expect RWR to always be lower than CA.

In the event when a model obtains desirable values for each metric, it is fruitful to perform more granular analysis, inspecting model performances for the various RWR-n metrics, individual composition rules and ground-truth answers in addition to qualitative analysis.

7.3. Experiments

In this section, we first describe the question types that we ignored during evaluation due to poor human validation scores and then detail how we trained and evaluated models. Afterwards, we perform an experiment exploring the use of AGQA-Decomp as data augmentation and provide additional analyses for the Most Likely baseline. We finally give examples of error modes that appeared during qualitative analysis.

Banned Question types. When evaluating model performance on the questions, we ignore questions of types that did not achieve at least a 70% human validation score. The following types did not achieve this threshold.

- **Action Temporal Localization**: This question type contains open answer questions for action recognition such as "What were they doing after walking through the doorway?" Human annotators marked 55.00% of questions of this type as correct.
- **Object**: This question type contains open answer questions for objects such as "What were they opening?". It also includes such questions when they have a temporal localization phrase. Human annotators marked 62.16% of questions of this type as correct.

Future work could address limitations in the AGQA dataset in order to improve the accuracy of questions or create a more accurate subset of questions. This new version could then be used to evaluate all types of questions.

Training Details. Upon running initial experiments with the default configurations of HCRN, HME and PSAC's respective repositories, we found that HCRN and PSAC overfit our data. As such, we performed hyperparameter searches for learning rate and weight decay parameters and additionally incorporated new dropout layers for each model to improve regularization. HCRN's best performing run was trained with a learning rate of 0.00016, a weight decay of 0.0005, a dropout probability of 0.15 and a batch size of 32. HME's best performing run remained the default configuration with a learning rate of 0.001, a weight decay of 0.0, no new dropout layers and a batch size of 32. PSAC, finally, was trained with a learning rate of 0.003, a weight decay of $5 * 10^{-6}$, a dropout probability of 0.15 and a batch size of 32. We trained HCRN for 5 epochs (where each epoch performs 18 validation loops), HME for 32000 update steps (corresponding to 40 validation loops) and PSAC for 23 epochs. We began terminated training after the validation accuracy of each model had plateaued. HCRN, HME and PSAC achieved best validation accuracies of 46.48%, 42.492% and 43.69% at the point of evaluation.

Using AGQA-Decomp as Data Augmentation Another intuitive application of AGQA-Decomp is data augmentation for the original AGQA dataset [5]. The training data

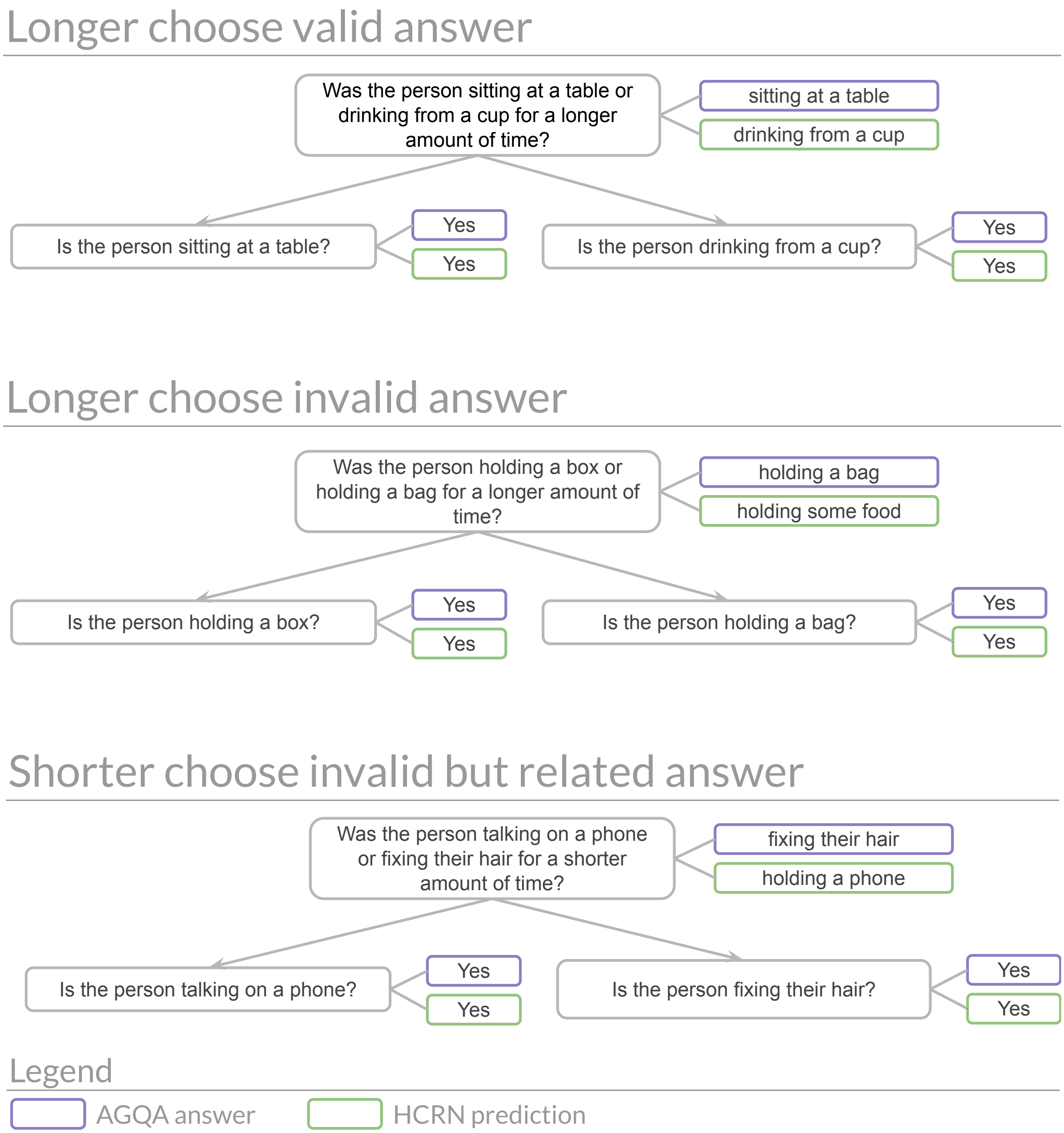

Longer choose valid answer
Was the person sitting at a table or drinking from a cup for a longer amount of time?
sitting at a table
drinking from a cup
Is the person sitting at a table?
Yes
Yes
Is the person drinking from a cup?
Yes
Yes
Longer choose invalid answer
Was the person holding a box or holding a bag for a longer amount of time?
holding a bag
holding some food
Is the person holding a box?
Yes
Yes
Is the person holding a bag?
Yes
Yes
Shorter choose invalid but related answer
Was the person talking on a phone or fixing their hair for a shorter amount of time?
fixing their hair
holding a phone
Is the person talking on a phone?
Yes
Yes
Is the person fixing their hair?
Yes
Yes
Legend
AGQA answer
HCRN prediction

Figure 11. We present example compositions where HCRN answers all children correctly but answers the parent incorrectly. **Top:** HCRN picks a valid but inaccurate option. **Center:** HCRN gives an unrelated response. **Bottom:** HCRN produces an invalid but relevant answer.

we used for our main evaluation is a version of the AGQA balanced dataset augmented with a balanced subset of questions taken from our DAGs. We can therefore investigate whether our trained models' performances are better than those trained on the standard AGQA dataset. We compare the accuracies of the best performing runs for both sets of models and find that using the AGQA-Decomp subquestion data naively as data augmentation does not result in a clear improvement. HCRN trained on AGQA-Decomp outperforms its counterpart trained on AGQA by 1%, while

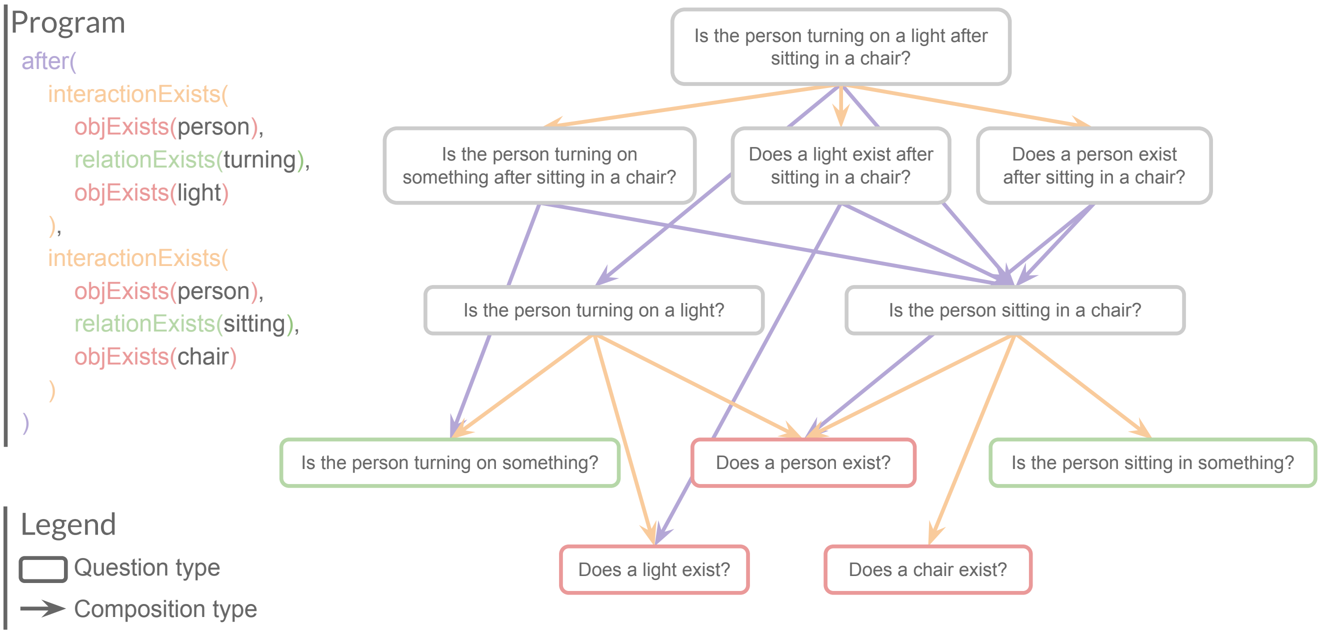

Program
after(
interactionExists(
objExists(person),
relationExists(turning),
objExists(light)
),
interactionExists(
objExists(person),
relationExists(sitting),
objExists(chair)
)
)
Legend
Question type
Composition type
Is the person turning on a light after sitting in a chair?
Is the person turning on something after sitting in a chair?
Does a light exist after sitting in a chair?
Does a person exist after sitting in a chair?
Is the person turning on a light?
Is the person sitting in a chair?
Is the person turning on something?
Does a person exist?
Is the person sitting in something?
Does a light exist?
Does a chair exist?

Figure 12. Example decomposition with corresponding program.

Table 6. We present HCRN, HME and PSAC performances on the **RWR-n** metrics, where n represents the exact number of incorrectly answered sub-questions for a composition, while conditioning on parent question types. Models are frequently accurate on parent questions even when answering simpler sub-questions incorrectly. For `Equals` and particularly `Interaction Temporal Localization` questions, **RWR-n** values largely outperform **CA** scores

	HCRN					HME					PSAC				
Parent Type	RWR-1	RWR-2	RWR-3	RWR-4	RWR-5	RWR-1	RWR-2	RWR-3	RWR-4	RWR-5	RWR-1	RWR-2	RWR-3	RWR-4	RWR-5
`Object Exists`	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
`Relation Exists`	16.67	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	20.22	N/A	N/A	N/A	N/A
`Interaction`	44.26	29.40	19.55	N/A	N/A	26.63	12.15	N/A	N/A	N/A	63.19	49.14	26.78	N/A	N/A
`Interaction Temporal Loc.`	49.23	55.34	56.26	39.65	6.25	89.05	93.92	70.33	71.69	4.35	29.37	58.69	65.44	28.15	9.12
`Exists Temporal Loc.`	67.93	21.98	N/A	N/A	N/A	2.25	1.83	N/A	N/A	N/A	30.23	4.52	N/A	N/A	N/A
`First/Last`	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
`Longest/Shortest Action`	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
`Conjunction`	48.17	31.21	N/A	N/A	N/A	47.42	27.85	N/A	N/A	N/A	46.17	30.60	N/A	N/A	N/A
`Choose`	47.51	41.12	N/A	N/A	N/A	48.57	39.74	N/A	N/A	N/A	48.24	47.32	N/A	N/A	N/A
`Equals`	52.10	50.20	N/A	N/A	N/A	47.08	47.54	N/A	N/A	N/A	51.19	47.69	N/A	N/A	N/A
Overall	55.59	32.24	47.31	39.65	6.25	30.05	13.43	70.33	71.69	4.35	40.75	27.73	57.47	28.15	9.12

HME, for instance, underperforms by 1% (Table 10). One possible reason for the lack of improvement is our use of sub-questions naively as more data. Future work may devise data augmentation schemes that go beyond this naive approach and leverage the structure provided by entire hierarchies for potentially better performance.

Further Comparisons with Most-Likely. We will provide further results and analyses involving the Most-Likely baseline in this section. The Most Likely baseline represents a model that relies primarily on linguistic biases, outputting the most likely answer for each basic question type. We will begin with a discussion of the Most-Likely baseline's IC results and then investigate individual question types and composition rules.

Performance on the IC metric: The Most-Likely baseline, on one hand, is perfectly consistent for one half of logical consistency rules, primarily the rules where the parent is answered "yes" and all child answers are also propagated to be "yes". On the other hand, it has no valid data points for the other half of the rules (Table 8). This is due to the fact that the Most-Likely baseline outputs the most common answer for each question type, severely restricting the parent-child answer distributions for each composition. Our overall IC metric avoids treating such biased models as highly consistent by performing a macro average of the consistency scores for each logical consistency rule associated with a question type or composition rule. Given that the Most-Likely baseline has undefined performances on a logical consistency rule for every single question type and composition rule, its IC values are treated as undefined on Tables 11 and 12.

Table 7. We present HCRN, HME and PSAC performances on the **RWR-n** metrics, where n represents the exact number of incorrectly answered sub-questions for a composition, while conditioning on composition rules between questions and their sub-questions. Models are frequently accurate on parent questions even when answering simpler sub-questions incorrectly. **RWR-1** and **RWR-2** scores reveal problematic reasoning for `And` and `Xor` compositions respectively for HME and PSAC.

	HCRN			HME			PSAC		
Compostion Type	RWR-1	RWR-2	RWR-3	RWR-1	RWR-2	RWR-3	RWR-1	RWR-2	RWR-3
`Interaction`	50.67	42.62	17.86	48.80	76.75	11.18	50.93	63.22	23.81
`First`	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
`Last`	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
`Equals`	52.10	50.20	N/A	47.08	47.54	N/A	51.19	47.69	N/A
`And`	48.35	15.40	N/A	79.38	6.32	N/A	80.11	10.61	N/A
`Xor`	48.03	51.99	N/A	24.95	86.73	N/A	22.54	82.81	N/A
`Choose`	47.72	42.20	N/A	48.63	36.76	N/A	48.47	47.59	N/A
`Longer Choose`	36.84	40.32	N/A	44.01	39.58	N/A	40.66	41.99	N/A
`Shorter Choose`	37.19	36.52	N/A	43.99	40.84	N/A	41.04	42.56	N/A
`After`	61.24	33.33	N/A	17.92	24.39	N/A	37.08	15.52	N/A
`Before`	65.16	36.90	N/A	17.40	23.90	N/A	34.81	15.77	N/A
`While`	66.01	21.34	N/A	10.09	8.94	N/A	32.36	10.20	N/A
`Between`	33.14	5.98	N/A	77.83	1.34	N/A	41.01	4.69	N/A
Overall	55.12	32.97	17.86	36.73	23.56	11.18	42.84	31.63	23.81

Table 8. We present internal consistency (**IC**) scores for individual logical consistency rules for HCRN, HME, PSAC and the Most-Likely baseline. Logical consistency rules being followed by "Yes" or "No" indicates that the parent question either is or implied to be "Yes" or "No". For `Choose` questions, "Object" and "Temporal" denote whether the parent is an object or "before" or "after." Models frequently achieve low values when the parent is "Yes" and are particularly inconsistent for `Choose` consistency rules.

Consistency Check	Parent Answer	IC			
		HCRN	HME	PSAC	Most-Likely
`Interaction`	Yes	75.70	27.30	0.00	100.00
	No	99.93	99.96	99.96	N/A
`Equals`	Yes	6.91	16.07	4.79	6.84
	No	88.21	86.00	90.44	N/A
`And`	Yes	74.37	57.96	5.89	N/A
	No	94.24	98.75	98.78	0.00
`Xor`	Yes	5.11	16.26	18.67	N/A
	No	82.29	88.47	94.00	100.00
`Choose`	Object	6.59	0.76	14.80	N/A
	Temporal	4.92	0.54	9.56	0.00
`After`	Yes	40.55	40.20	42.38	100.00
	No	99.93	99.97	99.97	N/A
`Before`	Yes	38.55	42.10	43.37	100.00
	No	99.92	99.99	99.98	N/A
`While`	Yes	42.64	33.91	44.70	100.00
	No	100.00	100.00	99.97	N/A
`Between`	Yes	88.87	42.01	79.00	100.00
	No	83.04	99.79	96.94	N/A
Overall		62.88	58.34	57.95	N/A

Performance on Choose and Equals: For the `Choose` question type, a category that contains a large set of possible answers, the Most-Likely baseline's performance is predictably poor with a CA score of 6.02% (Table 11). The

Table 9. We report accuracy per ground-truth answer for each binary question type expecting "Yes" or "No" answers for HCRN, HME, PSAC and the Most-Likely baseline. Models frequently perform well on one ground-truth answer at the expense of the other. HME particularly is biased towards "No" for all question types except `Object Exists`.

Question Type	Ground Truth	Accuracy			
		HCRN	HME	PSAC	Most-Likely
`Object Exists`	Yes	44.39	93.47	3.38	100.00
	No	49.70	0.00	86.67	0.00
`Relation Exists`	Yes	48.43	3.29	68.85	100.00
	No	55.84	99.11	4.02	0.00
`Interaction`	Yes	39.78	9.16	40.91	100.00
	No	53.65	91.98	83.76	0.00
`Interaction Temporal Loc.`	Yes	57.15	3.60	40.82	100.00
	No	41.91	97.24	49.58	0.00
`Exists Temporal Loc.`	Yes	59.42	1.32	28.58	100.00
	No	36.21	98.06	78.46	0.00
`Conjunction`	Yes	41.32	1.33	7.07	0.00
	No	57.88	98.82	92.94	100.00
`Equals`	Yes	41.91	1.88	19.80	100.00
	No	59.15	98.28	80.05	0.00

Table 10. We present the accuracy the best performing HCRN, HME and PSAC runs obtain when trained on the AGQA or the AGQA-Decomp balanced training sets. Models trained on AGQA-Decomp outperform those trained on AGQA, implying that our DAGs may potentially be useful sources of data augmentation. Accuracy for this table is the standard definition of accuracy.

	AGQA Accuracy		
Training dataset	HCRN	HME	PSAC
AGQA	42.11	39.89	40.18
AGQA-Decomp	**43.10**	38.96	39.75

Table 11. We report compositional accuracy (**CA**), right for the wrong reasons (**RWR**), delta (**RWR-CA**) and internal consistency (**IC**) metrics for the Most-Likely baseline with respect to question types. We find that whatever good performance the Most-Likely baseline achieves is within narrow slices of the dataset. N/A values under the IC column indicate that the model has no valid datapoints for at least one logical consistency rule for that question type.

	CA	RWR	Delta	IC
Question Type	Most-Likely	Most-Likely	Most-Likely	Most-Likely
Object Exists	N/A	N/A	N/A	N/A
Relation Exists	100.00	N/A	N/A	N/A
Interaction	79.00	87.61	8.61	N/A
Interaction Temporal Loc.	57.96	1.29	-56.67	N/A
Exists Temporal Loc.	98.79	97.58	-1.21	N/A
First/Last	N/A	N/A	N/A	N/A
Longest/Shortest Action	N/A	N/A	N/A	N/A
Conjunction	24.35	62.67	38.32	N/A
Choose	6.02	24.48	18.46	N/A
Equals	46.66	53.56	6.90	N/A
Overall	80.06	37.97	-42.09	N/A

Table 12. We report compositional accuracy (**CA**), right for the wrong reasons (**RWR**), delta (**RWR-CA**) and internal consistency (**IC**) metrics for the Most Likely baseline with respect to composition rules. We find that whatever good performance the Most-Likely baseline achieves is within narrow slices of the dataset, such as the case when parent and child questions are answered "No" and "Yes" respectively for Xor. N/A values under the IC column indicate that the model has no valid datapoints for at least one logical consistency rule.

	CA	RWR	Delta	IC
Composition Type	Most-Likely	Most-Likely	Most-Likely	Most-Likely
Interaction	64.07	48.91	-15.16	N/A
First	N/A	N/A	N/A	N/A
Last	N/A	N/A	N/A	N/A
Equals	46.66	53.56	6.90	N/A
And	0.00	100.00	100.00	N/A
Xor	100.00	40.39	-59.61	N/A
Choose	18.52	24.48	5.96	N/A
Longer Choose	5.73	N/A	N/A	N/A
Shorter Choose	6.22	N/A	N/A	N/A
After	79.91	53.15	-26.76	N/A
Before	80.51	54.22	-26.30	N/A
While	93.32	52.05	-41.26	N/A
Between	99.03	0.00	-99.03	N/A
Overall	75.60	37.70	-37.90	N/A

model only has valid datapoints for consistency checks on Choose compositions requiring choosing whether an event occurred before or after another. For this composition rule, the model is inconsistent for each case, as the child questions, which belong to the same question type, must be answered differently. Model performance on the Equals category is also poor, with the model being self-consistent only 6.84% of the time when the parent is answered "yes" (8)

Performance on Conjunction: For Conjunction questions, the Most-Likely baseline is biased towards "no" answers while it is biased towards "yes" answers for subquestions to Conjunction questions. As the Xor composition is always accurate for this answer distribution, the Most-Likely baseline obtains perfect CA score. Similarly, since the And composition is always inaccurate for this answer distribution, the model obtains 0.00% for CA. These extreme CA scores, the model's undefined IC values, as well as the high RWR score for And (Table 12) collectively indicate incorrect reasoning.

Performance on Temporal Reasoning: For temporal reasoning question types, such as Exists Temporal Localization and Interaction Temporal Localization, and their constituent composition rules (After, Before, While, Between), any good performance can be explained by the fact that the Most-Likely baseline answers only "yes" to both parent questions and its children. For these instances, the model is perfectly consistent. The IC scores being undefined on Tables 11 and 12 alert that the model does not reason compositionally yet again.

Qualitative Examples. In this section, we provide example illustrations of error modes we observed when models answered all immediate sub-questions questions correctly but answered the parent question incorrectly for the composition rule in which models achieved the worst performance: Longer and Shorter Choose. Figure 11 displays three error categories: one category where the model chooses the wrong option, one category where the model makes a semantically relevant prediction that is not given as an option and another category where the model makes a wholly irrelevant prediction.

7.4. Future work

While our analyses are limited to the AGQA benchmark, our decomposition structure can nonetheless facilitate multiple future contributions.

Consistency as a training loss Following in the path laid out by recent work [4, 10, 15], consistency can be operationalized as an additional training signal to encourage models to behave compositionally. The proliferation recent large language models [14] can be prompted to produce consistent training data augmentations for smaller models.

Interactive model inspection: Although the metrics that we propose each facilitate analyses across the entire dataset, they are motivated by how we expect models that reason compositionally should behave on individual examples. This makes the exploration of question DAGs as a tool for the interactive analysis of model behavior [13, 14] a fruitful direction.

Explanations through question decompositions: Furthermore, model answers to question hierarchies can be used as justifications of model predictions, similar to past work on natural language rationalizations [6, 8], with each answer representing model behavior in intermediate reasoning steps [3]. Internal consistency can similarly help determine whether to trust and rely on models.

This paper outlines several evaluation methods using a decomposition of AGQA questions. This application of a question decomposition structure already provides fruitful insights on model performance. The structure of AGQA-Decomp hierarchies can further provide both flexibility and nuance to evaluation outside of the use case explored here. We encourage future work to expand this structure to other benchmarks and to create novel evaluation methods.

References

[1] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021. 3

[2] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018. 3

[3] Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*, 2021. 12

[4] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Vqa-lol: Visual question answering under the lens of logic. In *European conference on computer vision*, pages 379–396. Springer, 2020. 12

[5] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3, 5, 8

[6] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European conference on computer vision*, pages 3–19. Springer, 2016. 12

[7] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787, 2018. 3

[8] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279, 2018. 12

[9] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020. 2, 3

[10] Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. Squinting at vqa models: Introspecting vqa models with sub-questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10003–10011, 2020. 12

[11] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 3

[12] Angelina Wang, Arvind Narayanan, and Olga Russakovsky. Revise: A tool for measuring and mitigating bias in visual datasets. In *European Conference on Computer Vision*, pages 733–751. Springer, 2020. 3

[13] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, 2019. 12

[14] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021. 12

[15] Yuanyuan Yuan, Shuai Wang, Mingyue Jiang, and Tsong Yueh Chen. Perception matters: Detecting perception failures of vqa models using metamorphic testing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16908–16917, 2021. 12

[16] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14830–14840, 2021. 3

[17] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei 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, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. 3