Supplementary material

A. Implementation

We provide below practical details of our implementation of the proposed method.

- Size of vector embeddings of node features, edge features, and all hidden states within the network: H=300. Note that smaller values such as H=200 also give very good results (not reported in this paper) at a fraction of the training time.
- Number of recurrent iterations to update graph node representations: T^{Q}=T^{S}=4. Anecdotally, we observed that processing the scene graph benefits from more iterations than the question graph, for which performance nearly saturates with 2 or more iterations. As reported in the ablative evaluation (Table 1), the use of at least a single iteration has a stronger influence than its exact number.
- All weights except word embeddings are initialized randomly following [10].
- Word embeddings are initialized with Glove vectors [20] of dimension 300 available publicly [18], trained for 6 billion words on Wikipedia and Gigaword. The word embeddings are fine-tuned with a learning rate of $1/10$ of the other weights.
- Dropout with ratio 0.3 is applied between the weighted sum over scene elements (Eq. 8) and the final classifier (Eq. 9).
- Weights are optimized with Adadelta [29] with mini-batches of 128 questions. We run optimization until convergence (typically 20 epochs on the "abstract scenes", 100 epochs on the "balanced" dataset) and report performance on the test set from the epoch with the highest performance on the validation set (measured by VQA score on the "abstract scenes" dataset, and accuracy over pairs on the "balanced" dataset).
- The edges between word nodes in the input question graph are labeled with the dependency labels identified by the Stanford parser [7, 19]. These dependencies are directed, and we supplement all of them with their symmetric, albeit tagged with a different set of labels. The output of the parser includes the propagation of conjunct dependencies (its default setting). This yields quite densely connected graphs.
- The input features of the object nodes are those directly available in the datasets. They represent: the object category (human, animal, small or large object) as one one-hot vector, the object type (table, sun, dog window, ...) as a one-hot vector, the expression/pose/type (various depictions being possible for each object type) as a one-hot vector, and 10 scalar values describing the pose of human figures (the X/Y position of arms, legs, and head relative to the torso). They form altogether a feature vector of dimension 159. The edge features between objects represent: the signed difference in their X/Y position, the inverse of their absolute difference in X/Y position, and their relative position on depth planes as +1 if closer (potentially occluding the other), -1 otherwise.
- All input features are normalized for zero mean and unit variance.
- When training for the "balanced" dataset, care is taken to keep each pair of complementary scenes in a same mini-batch when shuffling training instances. This has a noticeable effect on the stability of the optimization.
- In the open-ended setting, the output space is made of all answers that appear at least 5 times in the training set. These correspond to 623 possible answers, which cover 96% of the training questions.
- Our model was implemented in Matlab from scratch. Training takes in the order of 5 to 10 hours on one CPU, depending on the dataset and on the size H of the internal representations.

B. Additional details

Why do we choose to focus on abstract scenes ? Does this method extend to real images ?

The balanced dataset of abstract scenes was the only one allowing evaluation free from dataset biases. Abstract scenes also enabled removing confounding factors (visual recognition). It is not unreasonable to view the scene descriptions (provided with abstract scenes) as the output of a "perfect" vision system. The proposel model could be extended to real images by building graphs of the images where scene nodes are candidates from an object detection algorithm.

The multiple-choice (M.C.) setting should be easier than open-ended (O.E.). Therefore, why is the accuracy not better for binary and number questions in the M.C setting (rather than O.E.) ?

This intuition is incorrect in practice. The wording of binary and number questions ("*How many* ...") can easily narrow down the set of possible answers, whether evaluated in a M.C. or O.E. setting. One thus cannot qualify one as strictly easier than the other. Other factors can then influence the performance either way. Note also that, for example that most choices of number questions are not numbers.

In Table 1, why is there a large improvement of the metric over balanced pairs of scenes, but not of the metric over individual scenes ?

The metric over pairs is much harder to satisfy and should be regarded as more meaningful. The other metric (over scenes) essentially saturates at the same point between the two methods.

How are precison/recall curves helping better understand

model compared to a simple accuracy number ?
A P/R curve shows the confidence of the model in its answers. A practical VQA system will need to provide an indication of certainty, including the possibility of "I don't know". Reporting P/R is a step in that direction. P/R curves also contain more information and can show differences between methods (*e.g*. Fig.3 left) that may otherwise not be appreciable through an aggregate metric.

Why is attention computed with pre-GRU node features ?
This performed slightly better than the alternative. The intuition is that the identity of each node is sufficient, and the context (transfered by the GRU from neighbouring nodes) is probably less useful to compute attention.

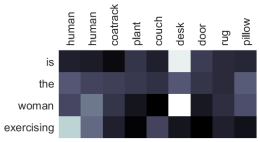
Why are the largest performance gains obtained with "number" questions ?
We could not draw definitive conclusions. Competing methods seem to rely on dataset biases (predominance of *2* and *3* as answers). Ours was developed (cross-validated) for the balanced dataset, which requires *not* to rely on such biases, and may simply be better at utilizing the input and not biases. This may in turn explain minimal gains on other questions, which could benefit from using biases (because of a larger pool of reasonable answers).

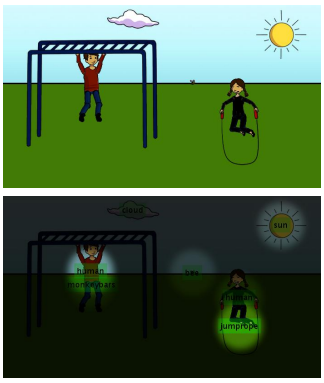
C. Additional results

We provide below additional example results in the same format as in Fig. 5.

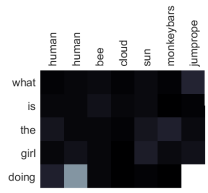
C.1. Additional results: abstract scenes dataset

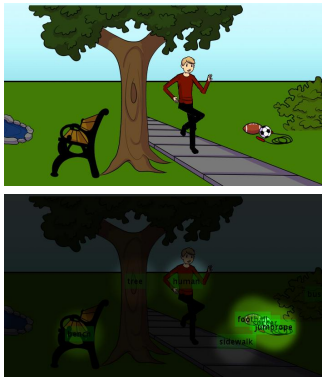
Is the woman exercising ?
Answer: yes



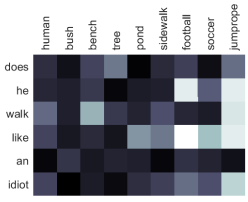


What is the girl doing ?
Answer: ~~jumping~~ jumping rope

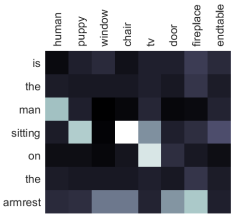


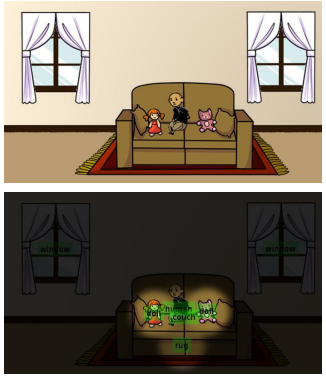


Does he walk like an idiot ?
Answer: yes

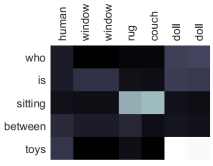


Is the man sitting on the armrest ?
Answer: ~~no~~ yes

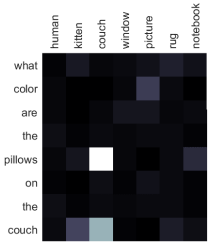


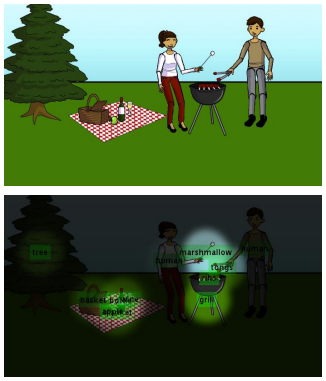


Who is sitting between toys ?
Answer: baby

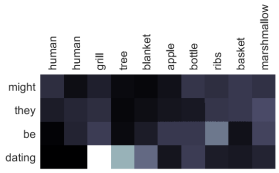


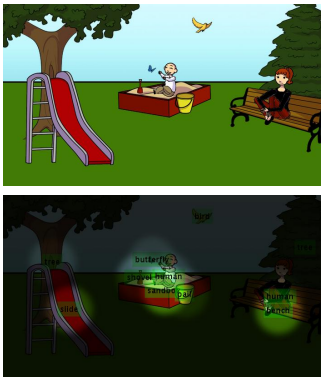
What color are the pillows on the couch ?
Answer: brown



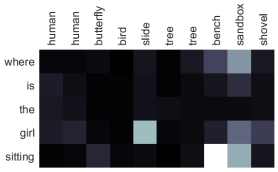


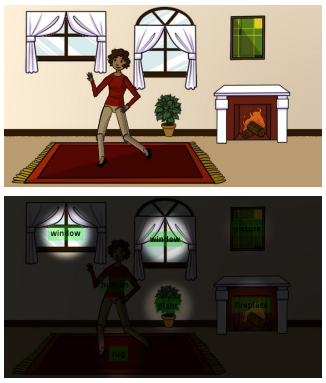
Might they be dating ?
Answer: yes



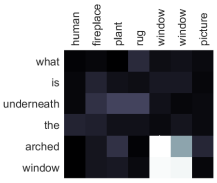


Where is the girl sitting ?
Answer: ~~sandbox~~ bench

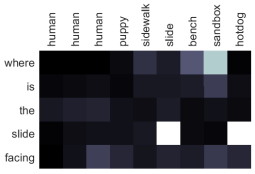


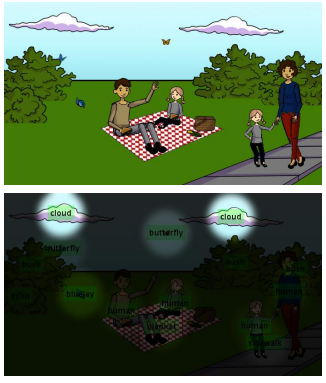


What is underneath the arched window ?
Answer: ~~rug~~ plant

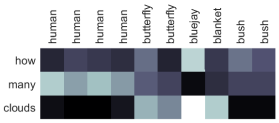


Where is the slide facing ?
Answer: ~~sandbox~~ left

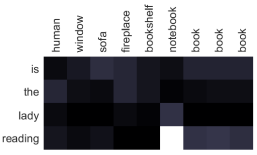




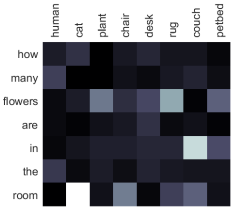
How many clouds ?
Answer: 2

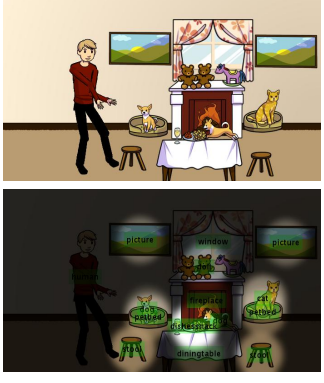


Is the lady reading ?
Answer: yes

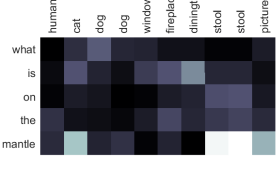


How many flowers are in the room ?
Answer: 0

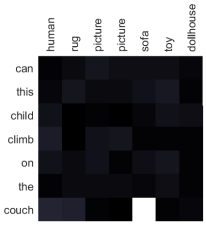




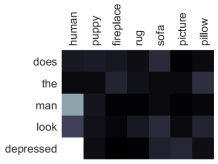
What is on the mantle ?
Answer: ~~wine~~ toys

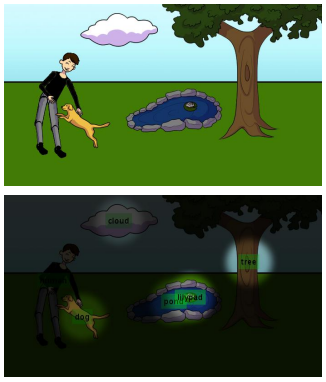


Can this child climb on the couch ?
no

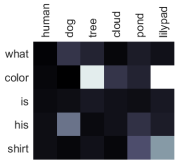


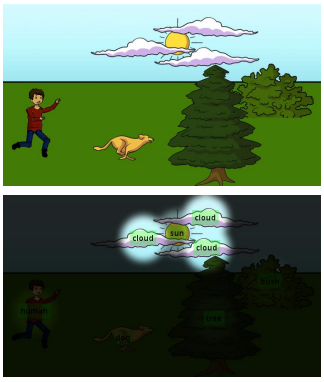
Does the man look depressed ?
Answer: yes



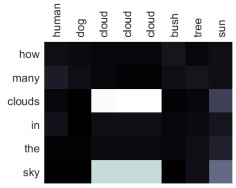


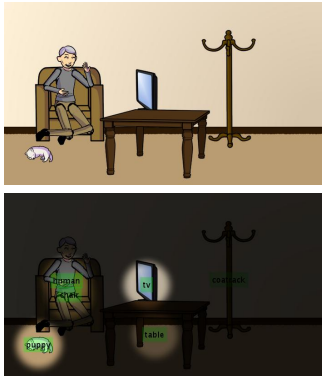
What color is his shirt ?
Answer: ~~red~~ black



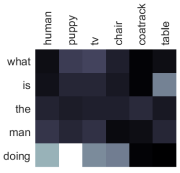


How many clouds in the sky ?
Answer: 3

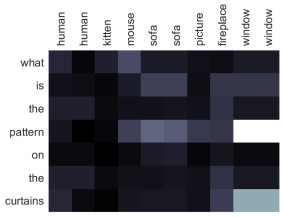


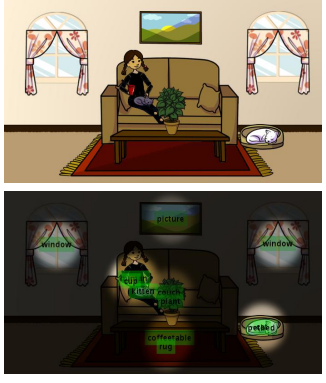


What is the man doing ?
Answer: ~~sitting~~ watching tv

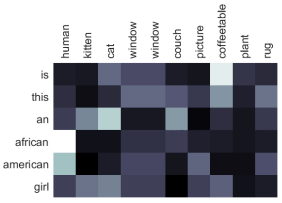


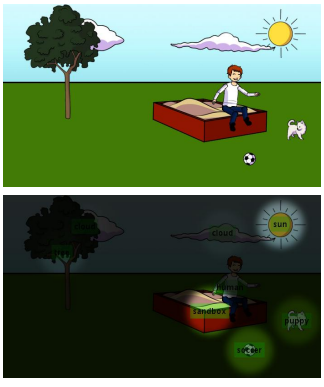
What is the pattern on the curtains ?
floral



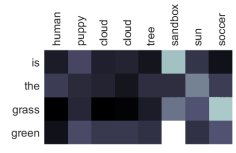


Is this an african american girl ?
Answer: ~~yes~~ no

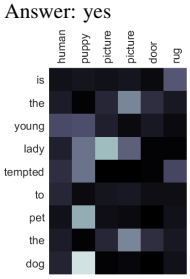




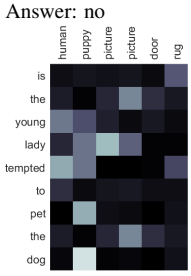
Is the grass green ?
Answer: yes



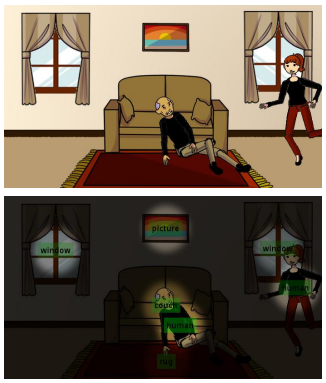
C.2. Additional results: balanced dataset

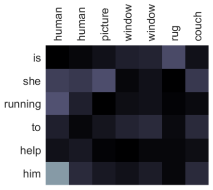


Is the young lady tempted to pet the dog ?
Answer: yes

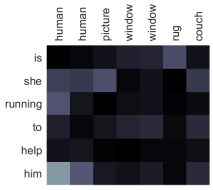


Is the young lady tempted to pet the dog ?
Answer: no

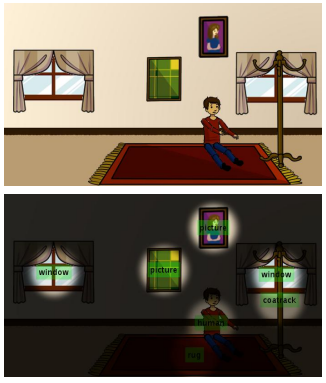


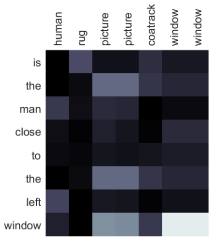


Is she running to help him ?
Answer: yes

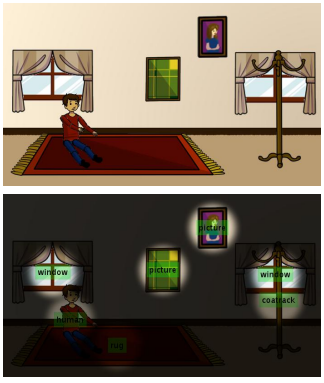


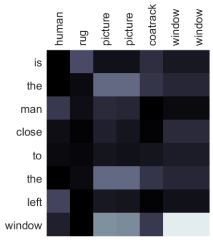
Is she running to help him ?
Answer: ~~yes~~ no





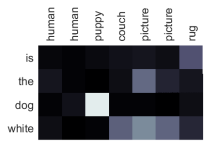
Is the man close to the left window ?
Answer: ~~yes~~ no



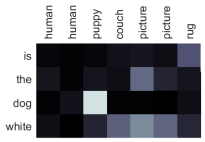


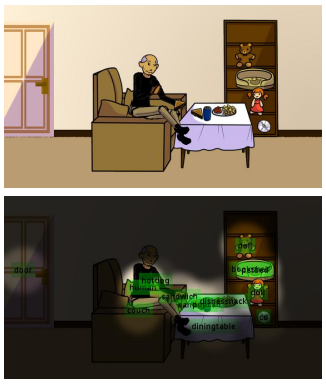
Is the man close to the left window ?
Answer: ~~no~~ yes

Is the dog white ?
Answer: yes

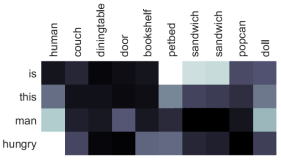


Is the dog white ?
Answer: no

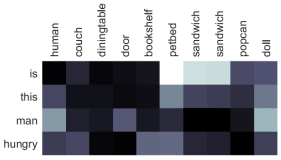




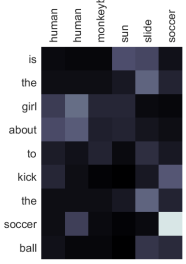
Is this man hungry ?
Answer: ~~no~~ yes

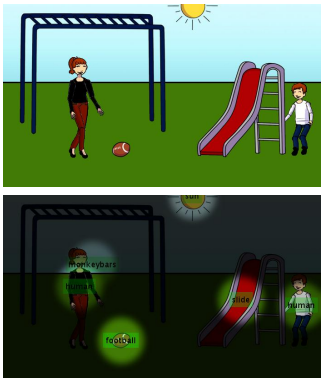


Is this man hungry ?
Answer: ~~yes~~ no



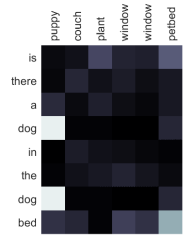
Is the girl about to kick the soccer ball ?
Answer: ~~no~~ yes



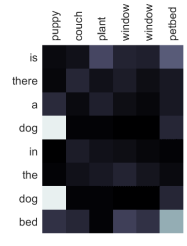


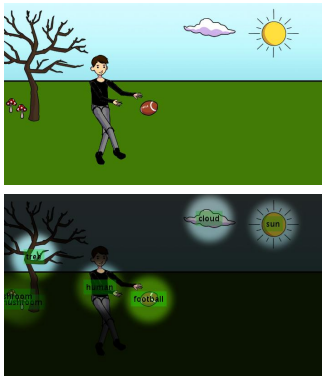
Is the girl about to kick the soccer ball ?
Answer: no

Is there a dog in the dog bed ?
Answer: ~~no~~ yes

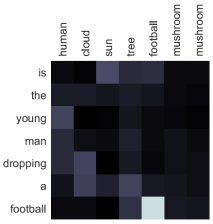


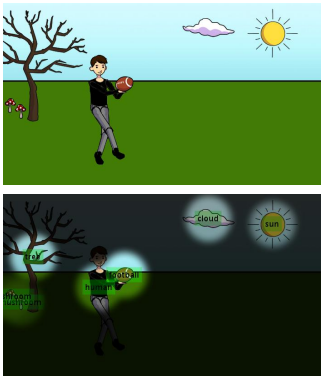
Is there a dog in the dog bed ?
Answer: no



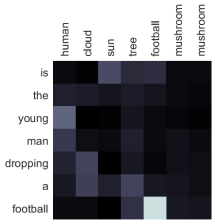


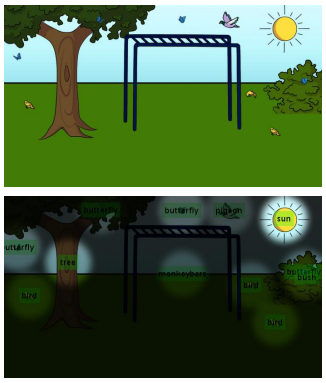
Is the young man dropping a football ?
Answer: yes



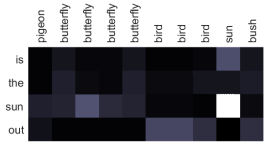


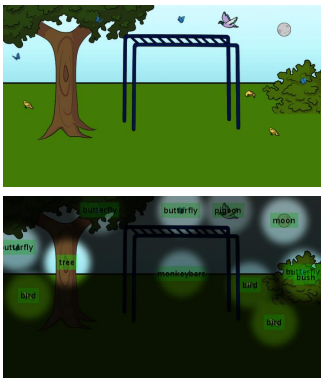
Is the young man dropping a football ?
Answer: ~~yes~~ no



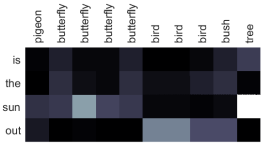


Is the sun out ?
Answer: yes

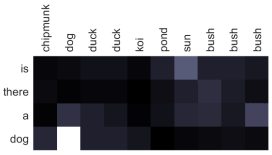


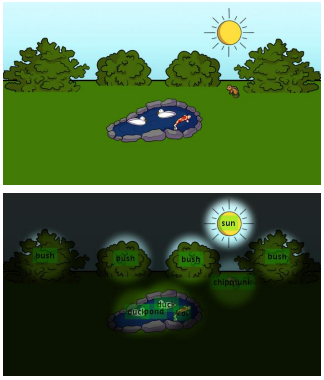


Is the sun out ?
Answer: no

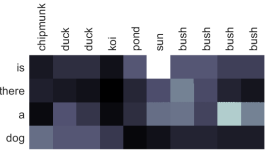


Is there a dog ?
Answer: yes

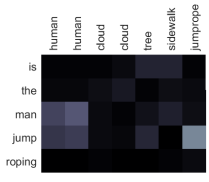




Is there a dog ?
Answer: no



Is the man jump roping ?
Answer: ~~no~~ yes



Is the man jump roping ?
Answer: no