Supplementary Material
Cycle-Consistency for Robust Visual Question Answering

The supplementary material is organized as follows:

- Section 1 covers information about the dataset collection pipeline, user interface and provides some dataset statistics.
- Section 2 shows qualitative examples of how attention over image regions varies for VQA models when different rephrasings of the same question are used as input.
- Section 3 describes an attention based consistency strategy that we experimented with, but did not improve performance (and so was not a part of our final model presented in the paper).
- Section 4 shows qualitative examples of answer conditioned questions generated by our VQG module.
- Section 5 lists the hyperparameters used for each base VQA model.
- Section 6 presents extended analysis of our failure prediction module.

1. Dataset Details

Statistics. Fig 1(a) shows the number of words (in percentage) belonging to different Parts-of-Speech tags. The distributions follow almost similar trends in VQA-Rephrasings and VQA v2.0. This shows that the rephrasings are not obtained by merely adding more adjectives or adverbs in the original question. Fig 1(b) shows the number of questions (in percentage) with varying lengths. The average length of questions in VQA-Rephrasings is 7.15 which is slightly higher than the average length in VQA v2.0, which is 6.32.

Interface. Fig 2 shows the interface used to collect rephrasings from human annotators. The interface provides three examples of invalid rephrasings and their corresponding explanations to help human annotators understand the task better. We A/B tested with 50 questions using all 4 combinations of:

- Showing both valid and invalid rephrasing examples and explanations.
- Showing only valid and no invalid rephrasing examples and explanations.
- Showing none of valid and invalid rephrasing examples and explanations.
- Showing no valid and only invalid rephrasing examples and explanations.

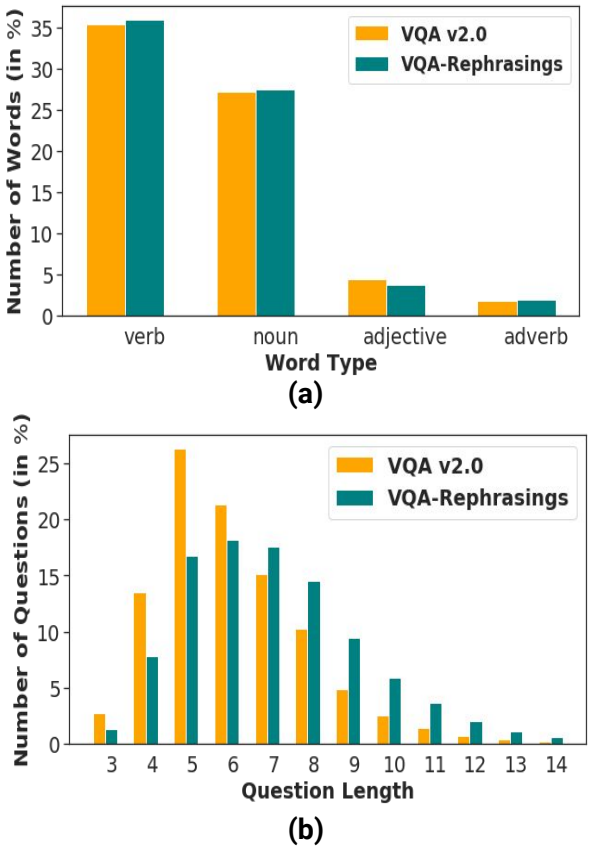


Figure 1. **Dataset Statistics.** (a) Shows the number of words (in percentage) belonging to different Parts-of-Speech tags. The distributions follow similar trends in VQA-Rephrasings and VQA v2.0. (b) Shows the number of questions (in percentage) with varying lengths. The average length of questions in VQA-Rephrasings is 7.15 which is slightly higher than the average length in VQA v2.0, which is 6.32.

We found (via manual inspection) that the last setup provided higher quality data, and used that as our final interface.

Examples. Fig 3 shows several qualitative examples from the VQA-Rephrasings dataset. We see that the rephrasings maintain the intent of the original question while varying linguistically.

Rephrase the question

Shown below is a question-answer pair about an image. The image is not shown. Rephrase or paraphrase the question in three different ways such that the meaning of the question remains the same. this requires that the answer to the rephrased question would remain the same.

You can rephrase it by making the question more to the point / succinct, by replacing common words by synonyms, by changing the order or anything else you can think of. You can also think about how you would have phrased the question yourself if you had asked it, and use that as one of the rephrasings. Below are two examples that are valid rephrasings of each other because the answer to the question would be the same.

IMPORTANT: As stated above, the meaning of the rephrased question should remain the same -- that is, the answer to the question should remain the same after you rephrase it. Below are two examples that are **NOT** valid rephrasings of each other because the answer to the question would not be the same.

INVALID REPHRASINGS

ORIGINAL QUESTION-ANSWER PAIR: **Question: "What color are the plastic utensils?" Answer: White**

- INVALID_REPHRASING: **"Do you think the plastic utensils are white in color?"**
  REASON: The answer to the original question would be a color, but in this case, the answer is either a yes or a no.
- INVALID_REPHRASING: **"Plastic utensils, what color?"**
  REASON: This is not a grammatically correct rephrasing
- INVALID_REPHRASING: **"What color are the utensils?"**
  REASON: The original question was asking about the color of the **plastic** utensils, but this rephrasing doesn't include it.

Note: Don't only think of ways to make the question longer. Also, consider ways to make the questions shorter. But overall, whatever comes naturally to you is fine as long as the above instructions are followed.

Rejection policy: You answer will be rejected if any of these are violated

- Make sure the submitted questions are valid grammatically correct rephrasings of the original question. See examples of invalid rephrasings above.
- Do not answer the question. The task is to rephrase the question, NOT answer it.
- Do not submit two or more exactly same rephrasings for a question.
- Do not submit the original question back as is.
- Do not submit very minor tweaks to the wording of the question. Provide real rephrasings.
- Do not leave a rephrasing blank.
- Do not use the exact same rephrasing strategies across rephrasings.

1st rephrasing of the question: ${question} | [Expected Answer: ${answer}]. Reminder: DO NOT ANSWER the question. Rephrase the question.

2nd rephrasing of the question: ${question} | [Expected Answer: ${answer}]. Reminder: DO NOT ANSWER the question. Rephrase the question.

3rd rephrasing of the question: ${question} | [Expected Answer: ${answer}]. Reminder: DO NOT ANSWER the question. Rephrase the question.

Figure 2. Interface used to collect question rephrasings given the original question (marked by $(question)) and ground truth answer (marked by $(answer))

2. Attention Analysis

Fig 4 qualitatively compares the textual and visual attention (over image regions) for rephrasings of a question. Each row compares predicted answers and attention from a baseline Pythia [4] model and the same Pythia model trained with our framework (Pythia + CC), using two question rephrasings. First and third row shows the outputs of a Pythia model (baseline) and second and forth row shows the output of a Pythia model (baseline + CC) trained with our framework. We see that in most examples, the attention over image regions doesn't vary across rephrasings for models trained with our framework (and the model answers the questions correctly). However for the baseline model, one can see that minor linguistic changes in the question can result in completely different answers (Row 2, Columns 1 and 3). This qualitatively demonstrates the robustness of models trained with our framework. Since the baseline Pythia model doesn't include a counting module, it doesn't perform well on questions requiring counting. As a result we see that both the baseline and its cycle-consistent counterpart perform poorly on counting questions (Row 5, Columns 1 through 4).

Model	Precision	Recall	F1
BUTD [1]	0.71	0.78	0.74
+ FP	**0.74**	**0.85**	**0.79**
BUTD + CC	0.73	0.79	0.76
+ FP	**0.78**	**0.83**	**0.80**
Pythia [4]	0.74	0.79	0.76
+ FP	**0.76**	**0.88**	**0.82**
Pythia + CC	0.77	0.81	0.77
+ FP	**0.82**	**0.84**	**0.83**

Table 1. **Failure prediction performance on VQA v2.0 validation dataset**. Each row in blocks represents a component added to the previous row. CC represents models trained with our cycle-consistent framework and FP represents models with an additional binary classification Failure Prediction module to predict if the predicted answer A' is correct given a question and image pair (Q, I). For models not using the FP module, failures are predicted by thresholding answer confidences. This is an extension of Table 4. in the main paper which highlights that models trained with cycle consistency are better at detecting failures both with and without an explicitly trained failure prediction module.

3. Attention Consistency

Intuitively, it seems like training the VQA model to attend over the same image regions for different rephrasings of a question should improve the robustness of the model. We tried to enforce this in our cycle-consistent framework using an additional attention consistency loss.

Recall that for a given image I, question Q and answer A, our model consists of a VQA model F which takes (Q, I) as an input and uses the question to attend over image regions with attention γ_Q and predicts an answer A'. We also have a VQG model G which uses the predicted answer A' and image I to generate a question Q'. Intuitively, the VQA model should attend over the same image regions when answering Q'. In other words, the attention over image regions $\gamma_{Q'}$ used by the VQA model to answer Q' should be close to the γ_Q. We added an additional attention consistency loss to the total loss which reduces the L_2 norm between these two attentions.

However, we found that this leads to reduction in model performance. Specifically, this reduces the performance of a cycle consistent Pythia model by 1.34% VQA accuracy when evaluated on the VQA v2.0 validation split (training on train split only).

We suspect one reason why enforcing attention consistency across rephrasings reduces performance is perhaps because minimizing a large number of diverse losses (cross entropy losses $\mathcal{L}_F$ and $\mathcal{L}_{cycle}$ for VQA, sequence generation loss $\mathcal{L}_G$ for VQG and mean squared loss $\mathcal{L}_{attention}$ for attention consistency) is a hard problem to optimize. Concretely identifying why enforcing attention consistency across question rephrasings hurts performance is currently under investigation and is part of future work. We find naively matching attentions across question rephrasings is not effective in current settings and therefore do not include this in the final model.

4. Question Generation

Fig 5 shows qualitative examples of answer conditioned questions generated by our VQG model. Our VQG model is able to correctly generate answer conditioned questions for a wide range of answers ranging from numbers, to colors and even yes/no.

5. Hyperparameters

We use the default hyperparameters as described in publicly available implementations of MUTAN [2], BUTD [1], Pythia [4] and BAN [3]. When using these models as base VQA models to train cycle consistent variants of them, we use the same parameters for the VQA model. For the the VQG model we use T_{sim}=0.9, λ_G=1.0, λ_C=0.5 and A_{iter}=5500. The hidden size of the LSTM used in VQG module is 1024 and the linear encoders used to encode the answer and image in VQG have dimensions of 300 each. While some models use adaptive learning rates for their base VQA models, the VQG model is always trained with a fixed learning rate of 0.0005. In case of BAN and Pythia, we also clip the gradients whose L_2 norm is greater than 0.25.

6. Failure Prediction

In the main paper, we show that by training models to generate and answer questions while being consistent across both tasks leads to improvement in performance and robustness. Another way of testing robustness of these models is to see if models can predict their own failures. As discussed in the main paper, we seek to verify if models trained with our cycle-consistent framework can identify their own failures *i.e*. correctly identify if they are wrong about a prediction. To this end, we use two failure predictions schemes. First, we naively threshold the confidence of the predicted answer. All answers above a particular threshold are marked as correctly answered and vice versa. Second, as described in the main paper, we design a failure prediction binary classification module (FP), which predicts for a given image I, question Q and answer A' (predicted by the base VQA model F), whether the predicted answer is correct for the given (I, Q) pair. This FP module uses image and answer encoders similar to those used in the question generation module and makes use of the question representation from the base VQA model as the question encoding. These encodings are concatenated and passed to a linear layer for binary classification. The FP module is trained keeping the parameters of the base VQA model frozen.

Table 1 is an extension of Table 4. in the main paper and shows that the cycle consistency framework, even *without* an explicit failure predictor module, makes the models more calibrated – more capable of detecting their own failures. As can be seen that in both settings: (a) when using naive confidence thresholding (not marked as "+ FP" in the Table) and (b) using a specifically designed submodule to detect failures (marked as "+ FP"), models trained with our cycle-consistent training framework are better than their corresponding baselines. We see similar trends in detecting failures for both BUTD and Pythia models, which shows that our cycle-consistency framework is model agnostic.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[2] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear Attention Networks. *arXiv preprint arXiv:1805.07932*, 2018.

[4] Yu Jiang*, Vivek Natarajan*, Xinlei Chen*, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

Figure 3. **Examples from our VQA-Rephrasings dataset**. The first question (shown in gray) in each block is the original question from VQA v2.0 validation set, the questions that follow (shown in black) are rephrasings collected in VQA-Rephrasings.

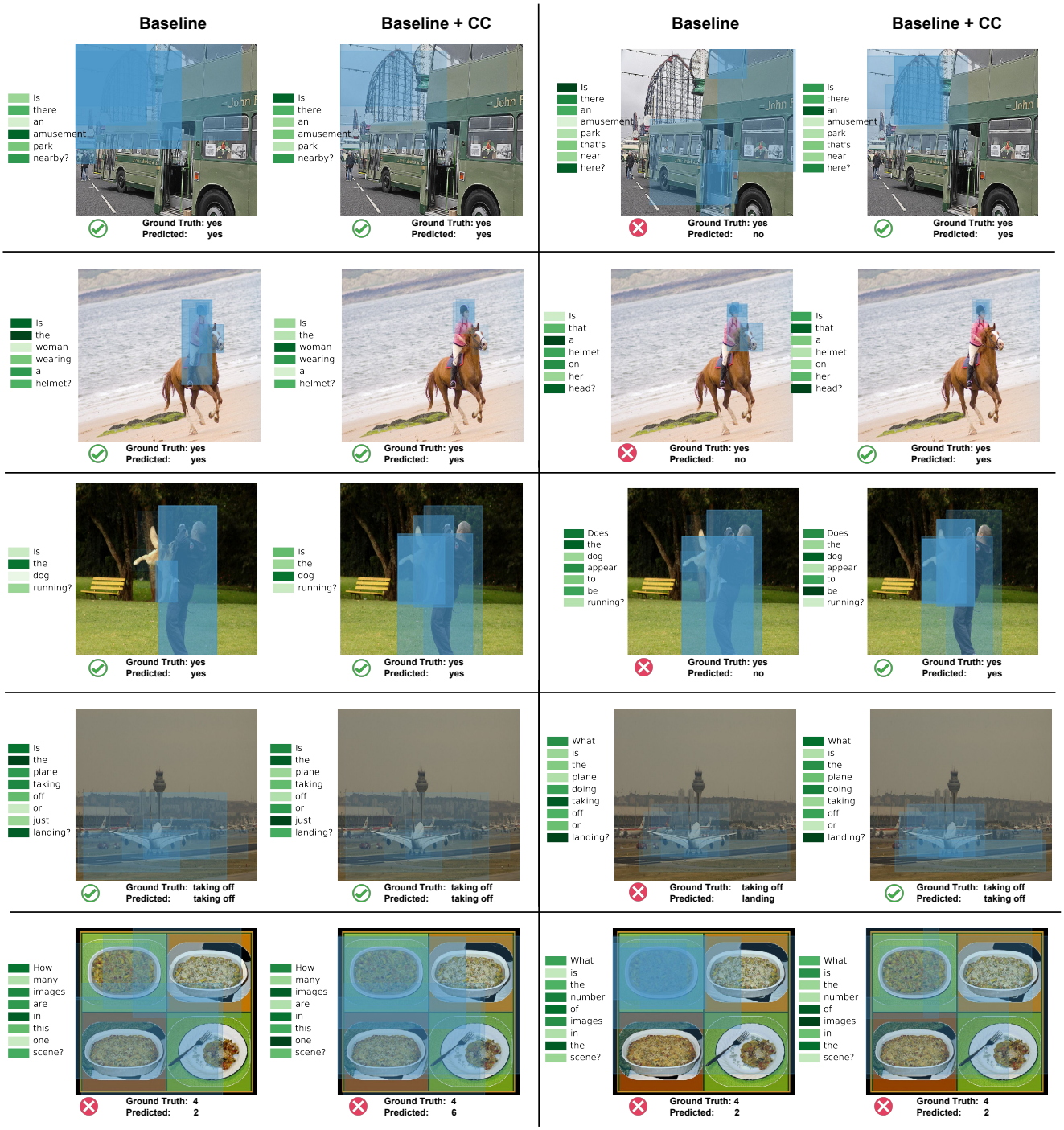
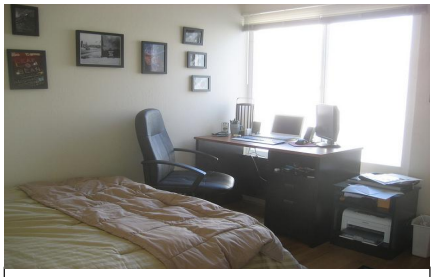


Figure 4. Visualization of textual and image region attention for different question variants: Each row compares answers predicted and attention for two question rephrasings using a baseline Pythia [4] model and the same Pythia model trained with our framework (Pythia + CC). Higher opaqueness in highlighted regions represents higher attention. First and third rows show the output of a Pythia model (baseline) and second and forth rows show the output of a Pythia model (baseline + CC) trained with our framework. As one can see, in most examples, the attention over image regions doesn't vary much for models trained with our framework. However for the baseline model, one can see that by very minor linguistic changes in the question it is possible to predict completely different answers (Row 2, Columns 1 and 3). These examples qualitatively demonstrate the robustness of models trained with our framework.



- One
  - How many chairs are in the room?
  - How many beds are present?
- No
  - Is this a hotel room?
  - Is there a person in the room?
- Yes
  - Is the bed made?
  - Is the bed white in color?
  - Is the wall white?
  - Does the desk look messy?

- Yes
  - Is this a winter a scene?
  - Are there trees in the background?
  - Is the man in air?
  - Is the man snowboarding?
- No
  - Is the man wearing googles?
  - Is there snow on the trees?
- Red
  - What color jacket is the man wearing?

- No
  - Is the bus moving?
  - Is the man in picture on the right side of the bus?
- Concrete
  - What is the sidewalk made of?
- Gray
  - What is the color of the bus?
  - What color is the vehicle in the picture?
- Daytime
  - Is it daytime or nighttime?

- Blue
  - What is the color of the suitcase?
- Red
  - What color is the door?
  - What is the color of the door?
- No
  - Is the man carrying a backpack?
- One
  - How many suitcases are there?
  - How many suitcases can be seen?

- Yes
  - Are there any chairs in the picture?
  - Are there flowers in the picture?
  - Are the flowers in a garden
- Seven
  - How many chairs are there in the picture?
- Grass
  - What is the object on the left?

- Apple
  - What fruit is shown in the picture?
  - What kind of fruit is that?
  - Which fruit is shown in the picture?
- Red
  - What color is the fruit shown?
  - What color are the apples?
- 10
  - How many birds can be seen in the picture?

Figure 5. Qualitative examples of answer conditioned question generation by our VQG module.