Separating Skills and Concepts for Novel Visual Question Answering (*Supplementary Materials*)

Spencer Whitehead[1], Hui Wu[2], Heng Ji[1], Rogerio Feris[2], Kate Saenko[2,3]
[1]UIUC [2]MIT-IBM Watson AI Lab, IBM Research [3]Boston University
{srw5,hengji}@illinois.edu, {wuhu,rsferis}@us.ibm.com, saenko@bu.edu

Appendix

A. Skill and Concept Details

To construct a comprehensive list of common skills required to answer a VQA question, we draw information from three sources: (1) our own annotation on 400 randomly selected VQA questions; (2) user study from [20]; and (3) previous work on question types [9, 10]. The user study in [20] only provides four types of vision skills. Existing work on question types have relevant information, however, the question types are not always directly translatable to our paradigm of skill and concept composition. For example, concept recognition is considered as a question type in [10] (object presence), but in our framework, it is considered as *concept grounding* rather than as a separate skill. Besides, existing question types are sometimes incomplete [10], or not representative of natural questions typically asked about images [9]. For instance, skills that require comparison or text reading form $\sim 6\%$ of the questions according to our labeling results, but they are not covered in [10]. We consolidate our annotations with groupings in existing work, which results in the following set of skills:

- Color recognition: *What color hair does the woman have? What color is his shirt?*
- Attribute recognition (non-color attributes): *Is the bed made? Is this desk messy?*
- Subcategory recognition: *What kind of car is parked? What kind of animals are shown?*
- Action recognition: *What is the man doing in the street? Are they comparing their phones?*
- Scene recognition: *Is this on a farm? Are they outside?*
- Counting: *How many lights are there? How many zebras are in this picture?*
- Commonsense knowledge: *Is the sun going down? Is this in America?*
- Positional reasoning: *What is on top of the toaster? What is the zebra standing on?*
- Text Recognition: *What number bus is it? What is the store called?*
- Comparison: *Is the tank the same color as the toilet? Are they facing the same direction?*

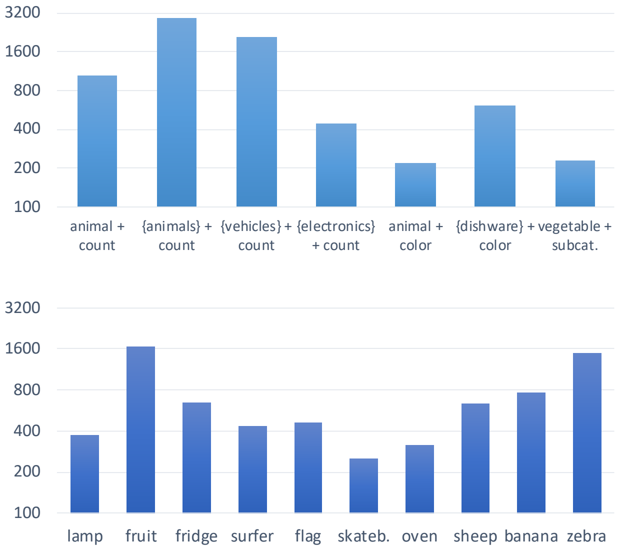

Figure 1. Novel skill-concept composition (top) and novel concept (bottom) question statistics.

We also provide additional information and statistics of the novel compositions. To facilitate further research on novel-VQA evaluation, we will provide concept and skill annotations, and the respective data indices for each set of novel composition. The list of concepts within each concept group is:

- {animals}: *giraffe*, *zebra*, *bird*, *sheep*, *horse*, *elephant*, *cow*, *dog*, *cat*
- {vehicles}: *motorcycle*, *airplane*, *plane*, *jet*, *bus*, *car*, *truck*, *bike*, *bicycle*
- {electronics}: *computer*, *monitor*, *laptop*, *phone*, *cellphone*
- {dishware}: *plate*, *bowl*

The list of sizes for each novel testing split is shown in Fig. 1. To determine the compositions/concepts that we use, we employ a few criteria: 1) each skill-concept composition (or concept) must have a minimum of 400 training questions and 200 testing questions; 2) for compositions, to increase coverage and ensure the minimally required size, we use concept groups where the concepts in a group all fall under a broader category (*e.g.*, {animals} = {*giraffe*, *zebra*,...}). We then sample from these compositions/concepts to conduct experiments on.

B. Approach Details

Here, we detail the projection functions, similarity functions, and other settings for our approach. In the following equations, all W and b are learned parameters.

Concept Grounding. For our concept grounding loss, we want to maximize the similarity of the masked target concept token to the correct concept token in the positive reference example. Since we are directly comparing tokens between examples, we model the similarity computation as an attention [3, 12, 18] with which the model must point [19] to the correct concept token. Specifically, our projection function, $\phi_g(\cdot)$, and similarity function, $\text{sim}(\cdot,\cdot)$, are defined as

$$\phi_g(x) = W_g x + b_g \tag{1}$$

$$\text{sim}(x,y) = \frac{x^\intercal y}{\sqrt{d}}, \tag{2}$$

where d is the dimension of x and y, $W_g \in \mathbb{R}^{d\times d}$, and $b_g \in \mathbb{R}^d$. Though this is similar to an attention, our formulation matches more traditional contrastive learning objectives [4, 13], where $\sqrt{d}$ is the temperature and we use a dot product as our similarity measure.

Skill Matching. Our skill matching loss seeks to maximize the similarity of the summary representation of the target question with the summary representations of other questions with the same skill. To obtain summary representations of questions, we simply use mean pooling over the question token representations. We define our projection function, $\phi_s(\cdot,\cdot)$, and similarity function, $\text{sim}(\cdot,\cdot)$, as

$$\phi_s(x) = W_s^{(2)}\psi(W_s^{(1)}x + b_s^{(1)}) + b_s^{(2)} \tag{3}$$

$$\text{sim}(x,y) = \frac{\cos(x,y)}{\tau_s}, \tag{4}$$

where ψ is a ReLU nonlinearity and τ_s is a temperature ($\tau_s = 0.5$ in our experiments). Since we are not directly comparing token representations, we use the more standard contrastive objective [4] as opposed to the attention-based formulation used for concept grounding.

Reference Sets and Training Procedure. When forming our CCC candidate references from which we sample our reference sets, we use $N^+ = 20$ and $N^- = 40$ (since we have two settings for negative examples), so there are N^+ positive and N^- negative examples that can be selected from to form a reference set for a given target example. Meanwhile, we use $N^+ = 200$ and $N^- = 200$ for our skill matching candidate references. Then, in our multi-task training procedure, we use $p_{\text{sep}} = 0.1$ as the probability of applying our framework at each training step. Additionally, we simply use $N_r^+ = 1$ and $N_r^- = 2$ for both concept grounding and skill matching, so the model will contrast between a single positive example and two distractor negative examples. For our concept grounding loss, we sample one negative example from both of our settings as our negative examples.

C. Experimental Details

C.1. Dataset Information

We use VQA v2 [7] for our main experiments. For training, we only use the training split. Since the testing data of VQA v2 [7] does not have public groundtruth information, we use the validation split of VQA v2 as the testing set for novel-VQA. To form our novel skill-concept and novel-concept VQA test splits, we automatically label questions with the skills and concepts using different NLP-based rules. For labeling skills, we use question template matching (*e.g.*, "*How many* ...") as well as verifying that the answers fit the matched templates. For labeling concepts, we utilize lemmatization and POS taggging and collect the frequent nouns. We then create different training splits that have a specific skill-concept composition or concept removed.

We also run experiments on the test-dev, test-std, and VQA-CP [1] splits of VQA v2. When evaluating on test-dev and test-std, we train on the validation set and additional Visual Genome data [17].

C.2. Model Configurations

All models use the same visual features [2].[1] We also use GloVe word embeddings [14].[2] Our baselines from prior work follow the recommended settings provided by the authors, whenever possible.

For XNM [15], we use the implementation provided by the authors as well as the recommended settings.[3] To ensure consistency between the two compositional models, we implement StackNMN [8] within the same code base as XNM. Specifically, we match the controller and the modules of StackNMN to the original paper. We use hidden dimension sizes of 512 for StackNMN and 1024 for XNM. We use the recommended number of reasoning steps, T = 3, for

[1] https://github.com/peteanderson80/bottom-up-attention

[2] Common Crawl 840B: https://nlp.stanford.edu/projects/glove/

[3] https://github.com/shijx12/XNM-Net

XNM and use the same for StackNMN. Both these models are trained with the Adam optimizer [11] and have the same learning rate of 0.0008 and batch size of 256.

For both X-Att [16][4] and X-BERT [5][5], we use the original model source code. For fair comparison, we do not use large-scale pre-training, same as our model. For X-Att, we use the recommended settings with a hidden size of 768, 12 layers, and 12 attention heads. X-Att uses the recommended learning rate of 0.0001, batch size of 64, 20 training epochs, and the Adam optimizer [11]. Due to their similarities in architecture, we use the same settings for Base, X-Bert and our framework for a more head-to-head comparison. Specifically, we use a hidden size of 512, 6 layers, and 8 attention heads. We match the training settings as well: a learning rate of 0.0001, batch size of 64, 13 training epochs, step learning rate decay with a rate of 0.2, and the Adam optimizer [11].

D. Base Model Architecture

The base model (Base), to which we apply our framework, is based on the standard transformer encoder [5, 6] with a few modifications. As is standard with transformers, we input visual regions, question tokens, and a special CLS that is appended to the beginning of the inputs, which we use to predict answers via a softmax output layer. There are two minor differences between a standard transformer and our base model: First, before inputting the question into the tranformer layers, we encode sequential information in the question tokens using an bi-directional LSTM, yielding a slight improvement than positional embeddings [18]. Second, in each layer, the CLS token and visual regions can attend to all inputs, including themselves, and the question tokens only directly attend to themselves and the CLS token. The change allows the CLS token to act as a bottleneck through which textual information interacts with the visual information.

E. Qualitative Examples

We show VQA output examples in Fig. 2 that compare the performance of our approach versus Base, where the first two rows show predictions on novel skill-concept compositions and the last row shows predictions on novel concept VQA. As a reminder, the models tested here never see labeled image-question pairs during training. Our approach allows the model to adapt to these unseen compositions. We see that, for unseen compositions of *counting* and different concepts, the base model struggles to recognize and count these concepts. For example, we observe that despite the clear appearance of the animals in the images, the Base model is unable to transfer the skill of counting, whereas the model trained with our framework is able to handle these cases. Similarly, in the third and fourth examples of the first row, we see an interesting effect where our approach is able to more precisely locate the specific "*plate*" being referred to. Another interesting example of the improvements that our grounding framework can offer is shown in the first three examples of the last row, where our model is able to locate the specific object and produce the correct answer. The last two examples of the third row show some intriguing failure cases, where our model produces plausible yet somewhat generic answers compared to the baseline.

[4] https://github.com/airsplay/lxmert

[5] https://github.com/ChenRocks/UNITER

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018. 2

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. 3

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 3

[7] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 2

[8] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *ECCV*, 2018. 2

[9] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 1

[10] Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. In *ICCV*, 2017. 1

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[12] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015. 2

[13] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

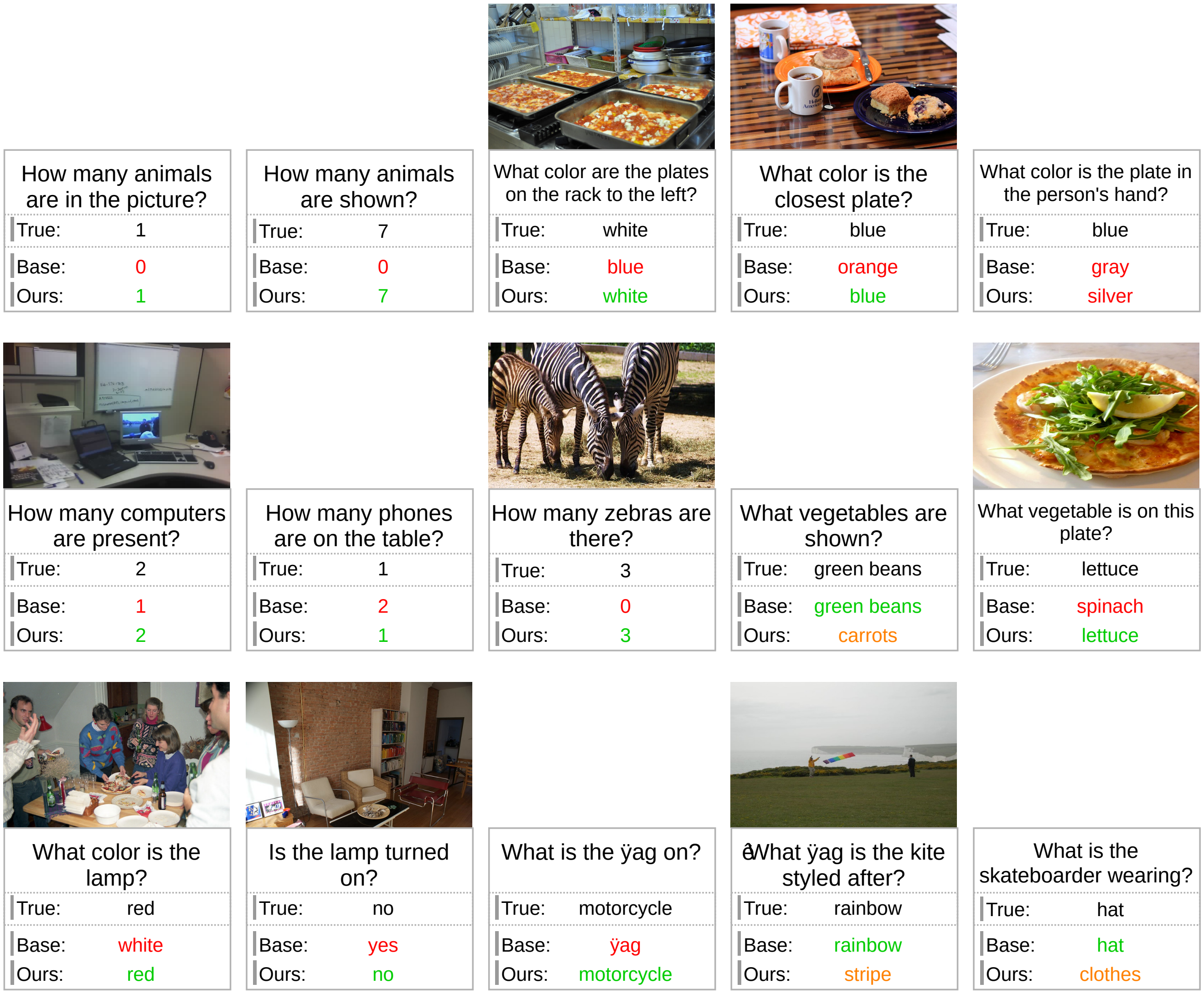

Figure 2. Correct, incorrect, and plausible VQA output examples for novel skill-concept composition VQA (rows 1 and 2) and novel concept VQA (row 3), comparing the predictions of our approach (Ours) and the Base model.

[14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2

[15] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, 2019. 2

[16] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 3

[17] Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*, 2018. 2

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[19] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NeurIPS*, 2015. 2

[20] Xiaoyu Zeng, Yanan Wang, Tai-Yin Chiu, Nilavra Bhattacharya, and Danna Gurari. Vision skills needed to answer visual questions. *Proc. ACM Hum.-Comput. Interact.*, 2020. 1