Supplementary: Weakly Supervised Grounding for VQA in Vision-Language Transformers

Aisha Urooj Khan[1], Hilde Kuehne[2,3], Chuang Gan[3], Niels Da Vitoria Lobo[1], and Mubarak Shah[1]

[1] University of Central Florida, Orlando, FL, USA
[2] Goethe University Frankfurt, Frankfurt, Hesse, Germany
[3] MIT-IBM Watson AI Lab, Cambridge, MA, USA

In this supplementary document, we further discuss about the proposed work as follows:

- Additional implementation details (section 1)
- Additional architecture details (section 2)
- Training objectives' details (section 3)
- Grounding Performance Evaluation (section 4)
- Performance analysis w.r.t. varying detection threshold (section 5)
- Grounding accuracy w.r.t. each head (section 6)
- Results w.r.t. question type (section 7)
- Entities represented by visual capsules (section 8)
- Training parameters in our method (section 9)
- VQA accuracy comparison (section 10)
- Additional details about evaluation on VQA-HAT (section 11)
- Qualitative Results (section 12)

1 Additional Implementation details

Here, we discuss the additional design choices used in our model. The capsules use routing to agree or disagree about the presence of certain entities in the input image. This decision is independent of the question. For instance, if an image has a *bus* and an *elephant* in it, the question cannot affect what is in the image. To this end, the capsule routing is performed only once in our method. However, depending on the question being asked, we may require selecting different entities from the image. We achieve this by doing text-based capsule selection at each layer. The capsule selection layer ϕ (equation 3 in the main paper) has shared weights for all encoder layers. We use 8 16GB AMD GPUs for pretraining; finetuning for GQA is performed on a single 16GB GPU. To pretrain with 48 capsules, the batch size of 640 is used.

2 Additional Architectural details

Language Encoder: The language encoder L_e is composed of L transformer encoder layers. Its input is a tokenized sentence S_l of length l. The language

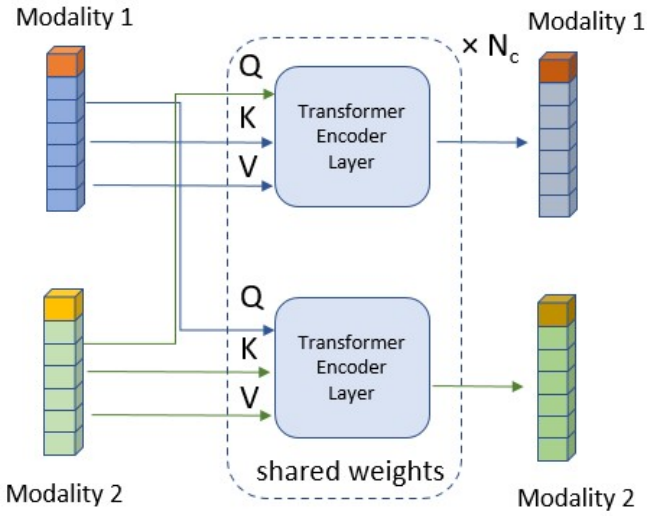


Fig. 1: Cross-attentional module used in our architecture. We use two cross-attentional layers, i.e., $N_c = 2$ in our best model.

encoder L_e takes the set of words tokens $\{[CLS], w_1, w_2, ..., w_l, [SEP]\}$ as input, and outputs feature representations $\{h^i_{cls}, h^i_{T_1}, h^i_{T_2}, ..., h^i_{T_l}, h^i_{sep}\}$ at every i^{th} layer, where $h^i_{T_k}$ denotes the text feature for the k^{th} input word token ($k \in 1, 2, ..., l$) from layer i. Intermediate encoder layer i takes output of previous layer $(i-1)$ as input. $[CLS]$ token is used as the sentence embedding in transformers [4]. Additionally, we use it for capsule selection in visual encoder (section 3.3 in main).

Visual Encoder: The visual encoder V_e has the same architecture as the language encoder with the same dimension size and number of layers. The image embeddings X' are transformed to visual capsules encodings X_c and input to the visual encoder. Intermediate layers of visual encoder takes selected visual capsules as residual connection to keep the capsule representation intact while training the system. The final features output $h^L_{v_j}$ of the visual encoder is used for token-level cross-modality interactions in future steps. Where, $h^L_{v_j}$ is the feature output for j^{th} visual token ($j \in 1, 2, ..., hw$) from the last layer L.

Feature Pooling The feature pooling layer takes text-based features and image-based features as input and outputs a d dimensional feature. This output feature can be used as a pooled output for image-text matching and VQA tasks. The feature pooling layer is a fully connected layer followed by a *tanh* activation layer. We pretrain our system in two stages before finetuning for VQA task. To be specific, during first stage pretraining , the input is the concatenated features $[h^L_{cls}, h^L_{img}]$ for special tokens from text and image encoders; where h^L_{cls}, and h^L_{img} are used as aggregated features over text input and image input respectively. Let f_P be the feature pooling layer, the pooled feature output $h_{1_{pooled}}$ will be as

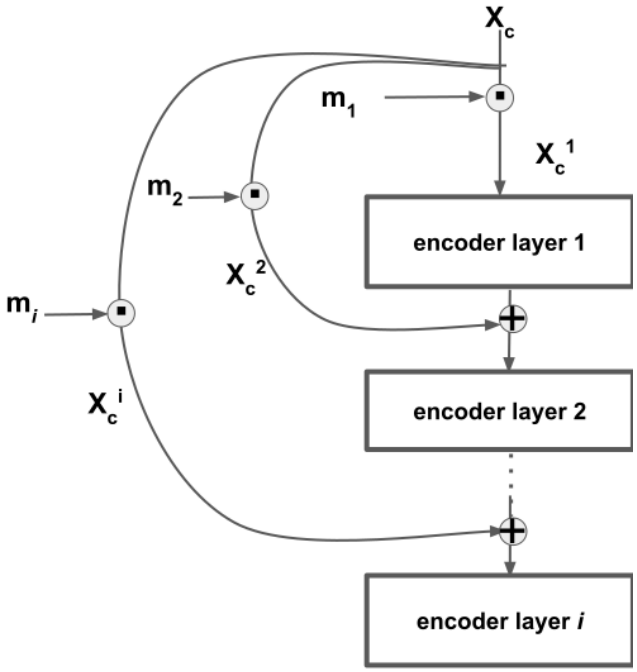


Fig. 2: A Simplified illustration of residual connections from capsules to transformer layer's input. Notation used from the main paper.

follows:

$$h_{1_{pooled}} = f_P([h^L_{cls}, h^L_{img}]) \tag{1}$$

During second stage pretraining, the concatenated features pair after cross attention is indicated as $[h^{\hat{L}}_{cls}, h^{\hat{L}}_{img}]$ and the pooled feature output is denoted by $h_{2_{pooled}}$. The equation is as follows:

$$h_{2_{pooled}} = f_P(h^{\hat{L}}_{cls}, h^{\hat{L}}_{img}]) \tag{2}$$

Cross-Attention Module Given two input feature sequences (output $h^L_{T_k}$ from text encoder and $h^L_{v_j}$ from image encoder), cross-attention module is a co-attentional transformer which applies attention from one feature sequence to the other by taking queries from first sequence and keys and values from the second sequence, and vice versa. Multiple layers of these cross-attention blocks can be stacked. The final text output feature $h^{\hat{L}}_{cls}$ corresponding to the $[CLS]$ token and final visual output feature $h^{\hat{L}}_{img}$ corresponding to the $[IMG]$ token are used for pretraining and finetuning the model. Where, N_c is the number of layers in cross-attention module and $\hat{L} = L + N_c$ denotes the depth of the model in terms of number of layers.

3 Training objectives

Masked Language Modeling Masked Language Modeling is a self-supervised language modeling task where a small percentage of words are masked before giving the sentence as input to the language encoder. The task is to predict the masked words using the context from other words in the sentence. This self-supervised approach is very effective to learn strong text representations [4]. The features output $h^L_{T_k}$ from language encoder L_e is used for training on this task. In the second stage, instead of predicting missing words from solely text features, the masked word is predicted from the visual-guided language features i.e., we take features outputs $h^{\hat{L}}_k$ from the last text-based cross-attention layer.
Image-Text Matching (ITM) To predict whether the input pair of image-text features is a matching pair or not, we take the output $h_{1_{pooled}}$ (eq. 1) from feature pooling layer and input to a fully connected layer which outputs logits for each class: 'matching' or 'non-matching'. At the second pretraining stage, the output features corresponding to $[IMG]$ and $[CLS]$ token after cross-attentional module (each of dimension d) are used for prediction. The pretraining head now uses $h_{2_{pooled}}$ (eq. 2) as input for image-text matching task.
Visual Question Answering Inspired by [11], we also use VQA as one of our pretraining tasks. We use Visual7W [12], GQA [7] and VQA [5] in our pretraining. Like ITM, we take the pooled features from text and visual encoders and input to a classifier. The classifier is comprised of two fully connected layers. An activation function and layer norm is used between the two layers. The final output is probability scores for each answer. In the first stage of pretraining, $h_{1_{pooled}}$ is used as the pooled feature. For second stage pretraining, pooled cross-modal feature output $h_{2_{pooled}}$ is used for answer prediction. A separate softmax cross-entropy loss function is used to optimize each of the above heads. We give equal weights to each loss term during pretraining.
Finetuning parameters for the baselines. To finetune LXMERT, we use the same training parameters as our method. ViLT is finetuned with batch size of 256 with lr=$1e-5$. ALBEF is finetuned with batch size of 16. Learning rate is increased to $2e-5$ to speed up training for ALBEF. ALBEF and ViLT use the maximum batch size which could fit in the GPU memory. All models are trained on GQA for upto 10 epochs.

4 Grounding Performance Evaluation

Choice to use last layer's attention for grounding: It is common in the vision-language community to employ the last layer's analysis e.g., DINO [2] uses the last layer's attention for the object segmentation task without any specialized training objective or architecture. We follow the protocol of previous works in the field for SOTA comparison to allow for a fair evaluation, namely following MAC [6], MAC-Caps [8] with mean (last) attention scores, ALBEF [10] by using the 8th and last layer with Grad-CAM (GC) and attention scores (ATN) and ViLT [9] for the last layer cosine (cos) and attention scores (see Tab. 2 in the

	Overlap			IOU		
Obj. label	P	R	F1	P	R	F1
Answer (A)	17.64	89.81	29.49	2.05	10.45	3.42
Question (Q)	49.57	81.86	61.75	4.01	6.48	4.95

Table 1: GradCAM results for our model. Compare to Tab. 2 in the paper.

main paper) and achieve SOTA grounding performance. However, there is a possibility that some intermediate layer does the better job at grounding such as ALBEF finds that layer 8 in their model is good at grounding. Nevertheless, searching for the best layer is expensive in terms of time and computational cost. Our approach outperforms on grounding even when evaluated for the last layer only. This choice also eliminates the need to search for the best grounding layer within each model and well suited to test the systems for unseen data.

Ours + Grad-CAM: To compare with ALBEF, we also evaluated our system with Grad-CAM output of the last layer. We observe $\approx 2-4\% \uparrow$ increase in overlap F1-score for both Q & A and a ($\approx 0.5-1.01\% \downarrow$) decrease in IOU F1-score still achieving better grounding results than the baselines (see Tab. 1 and Tab. 2 (main)).

4.1 Additional Ablations

Language guidance through [CLS] token vs. all word tokens: [CLS] token represents a fixed dimensional vector representing the sentence feature in transformers. Thus, output embedding for [CLS] token already captures the attended words in each layer while the computation cost remains unaffected if the question length increases. To evaluate this point, we finetune our pretrained backbone on GQA with capsules' mask generation using all token embeddings. First, we reduce the tokens embedding size from 768 to 128 with an fc layer, then concatenate all tokens forming a feature vector of size 2560 (128×20 for 20 question words we used for GQA). We use 2 fc layers mapping dimension $2560 \rightarrow 768 \rightarrow 16$ for 16 capsules. Results are shown in row 1, Tab. 2.

Finetuning on GQA with stage 1 pre-training We report the results with first stage pretraining in row2, Tab. 2. We conclude that the cross-modal layer, trained in stage 2 is relevant to guide the overall grounding, as here visual and textual attentions attend each other. We assume that the cross-modal attention finally allows the capsules to learn which concepts are finally relevant for a specific answer, which also shows in the fact that without this layer, while VQA accuracy increases, the F1 score is significantly lower compared to the 2-stage pre-training (row 3, Tab. 2).

Method	Overlap				IOU		
	Acc.	P	R	F1	P	R	F1
(1) all tokens (C=16)	56.69	13.15	81.63	22.65	1.90	11.85	3.27
(2) [CLS] token (C=16)	**57.21**	**14.53**	**85.47**	**24.84**	**2.30**	**13.61**	**3.94**
(3) stage 1 pre-train. (C=16)	**59.68**	8.13	61.07	14.35	0.93	7.03	1.64
(4) stage 2 pre-train. (C=16)	57.21	**14.53**	**85.47**	**24.84**	**2.30**	**13.61**	**3.94**

Table 2: Results on GQA val set.

Generating heatmaps using ViLT demo: The demo code visualizes word-to-patch attention for the matching image-caption pair. For VQA grounding, we consider question-to-image attention (attention from [CLS] token to visual tokens). The provided code for optimal transport algorithm leads to numerical instability for the question token ([CLS]) generating NaN. Hence, we used the cosine scores (computed before optimal transport) as well as raw attention to generate heatmaps. Heatmaps are generated with the same post processing as provided in the demo. To verify this, ($< question\_id >$, $< image\_id >$) pairs for Fig.4 (main) are: {('00798998', '2356417'), ('02451905', '2386586'), ('00653991', '2324955'), ('00511505', '2410567'), ('01782610', '2409395') }.

			Overlap			IOU		
Q Type	Example	Method Acc.	P	R	F1	P	R	F1
Open	*How is the weather in the image?*	no-caps 62.43	23.03	46.02	30.70	4.83	9.66	6.44
		ours 57.21	**43.06**	**85.57**	**57.29**	**6.62**	**13.24**	**8.83**
Binary	*Is it cloudy today?*	no-caps 62.43	33.19	66.10	44.20	7.13	14.25	9.50
		ours 57.21	**42.54**	**84.67**	**56.63**	**12.12**	**24.23**	**16.15**
Query	*What kind of fruit is on the table?*	no-caps 62.43	33.19	66.10	44.20	4.83	9.66	6.44
		ours 57.21	**43.06**	**85.57**	**57.29**	**6.62**	**13.24**	**8.83**
Compare	*Who is taller, the boy or the girl?*	no-caps 62.43	9.96	19.91	13.27	1.11	2.21	1.47
		ours 57.21	**37.83**	**75.00**	**50.29**	**2.43**	**4.87**	**3.24**
Choose	*Is it sunny or cloudy?*	no-caps 62.43	24.62	49.19	32.82	7.86	15.72	10.48
		ours 57.21	**43.11**	**85.85**	**57.40**	**13.29**	**26.59**	**17.73**
Category	*What kind of fruit is it, an apple or a banana?*	no-caps 62.43	30.14	60.10	40.14	8.66	17.32	11.54
		ours 57.21	**42.90**	**85.49**	**57.13**	**11.73**	**23.46**	**15.64**
Relation	*Is there an apple on the black table?*	no-caps 62.43	33.45	66.60	44.53	4.16	8.31	5.54
		ours 57.21	**43.10**	**85.60**	**57.33**	**5.84**	**11.68**	**7.79**
Attribute	*what color is the apple?*	no-caps 62.43	9.96	19.91	13.27	1.11	2.21	1.47
		ours 57.21	**37.83**	**75.00**	**50.29**	**2.43**	**4.87**	**3.24**

Table 3: Comparison of our backbone model with no capsules (no-caps) and the proposed model with 16 capsules (Ours (C=16)). Results are shown w.r.t. each question type. Adding capsules to the backbone model significantly improves the grounding performance for all question types.

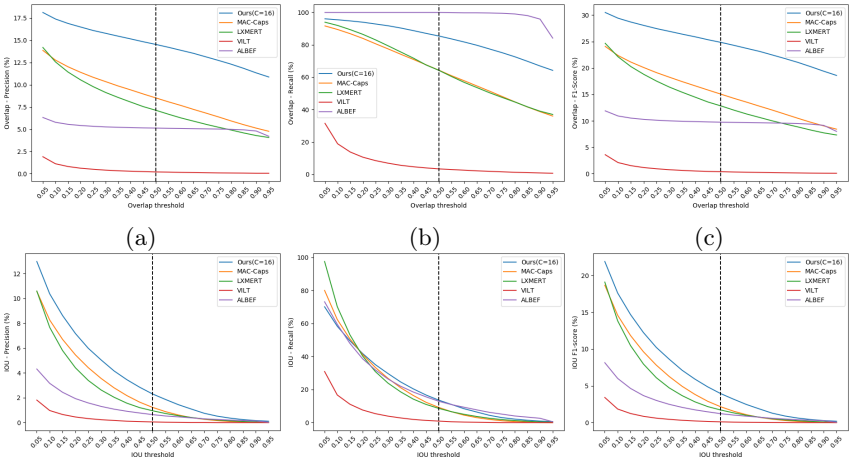


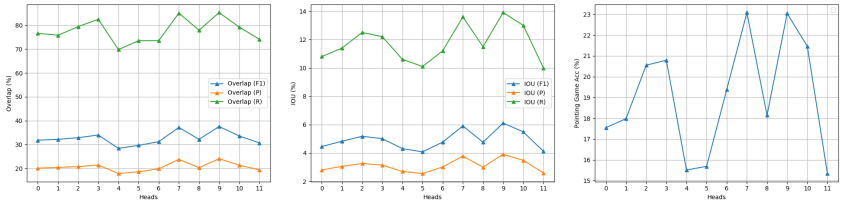
Fig. 3: Comparison with baselines for varying overlap and IOU detection threshold from 0.05 to 0.95. Plots (a), (b), and (c) show the results for varying detection threshold for overlap in terms of precision, recall, and F1-score respectively. Plots (d), (e), and (f) are the results for comparison when varying IOU threshold in terms of precision, recall, and F1-score respectively. Our method is significantly outperforming the baselines for all values of overlap thresholds in terms of precision, and subsequently F1-scores. For IOU, the proposed method is doing well for threshold values as high as 0.8 in terms of precision and F1-score, whereas, IOU-Recall is comparable to ALBEF.

5 Performance analysis w.r.t. varying detection threshold

We use detection threshold=0.5 for all our results in the submitted paper. For overlap, a detection is considered to be a true positive when the overlap between the ground truth box and the predicted region is greater than 0.5. Similarly, a detected region with an IOU of greater than 0.5 over a ground truth bounding box is considered a true positive for IOU. In figure 3, we study the impact of having a very low detection threshold vs. employing high thresholds by varying the threshold from 0.05 upto 0.95. We observe that the proposed method is robust to detection threshold for the overlap metric in terms of precision and F1-score even for the very high threshold of 0.95. For IOU, we also perform well for precision and F1-score. For IOU in terms of recall, our method and ALBEF show comparable results.

6 Grounding accuracy w.r.t. each head

Grounding accuracy for individual heads in the last cross-attentional layer are shown in figure 4. The results are reported in terms of precision, recall, and F1-score for overlap and IOU. For pointing game, the maximum point over the attention map produced from each head is used to evaluate the per-head



(a) (b) (c)

Fig. 4: Grounding performance from the proposed model (C=16) for each head in the last cross-attention layer. (a) reports overlap accuracies in terms of precision, recall, and F1-score; (b) shows IOU in terms of precision, recall, and F1-score; and (c) shows pointing game accuracy for each head. Overall, head 7 and head 10 show best grounding performance among all heads. For pointing game, head 7 achieves the highest accuracy of 23.08%. Using the proposed way to evaluate the pointing game performance, i.e., clustering over maximum points from all heads, improves pointing game accuracy significantly (34.59%).

pointing game accuracy. Using clustering over the points obtained from each head outperforms the best performing head by $\uparrow$ 11.51% (best head: 23.08% vs. clustering: 34.59%).

7 Results w.r.t. question type

Table 3 shows grounding results of our best model for different question types. GQA has questions classified with respect to structural type and semantic type. We compare our model with our backbone model which uses no capsules. Our system outperforms over all question types for both overlap and IOU particularly for question type "compare", "choose", and "attribute". Examples for each question type are provided in table 3.Refer to GQA [7] for more details about the question types present in this dataset.

8 Entities represented by capsules

Since, we do not use class labels to train the capsules, and use VQA supervision instead for training the whole system, it is hard to guess which entity is being represented by each capsule. To examine what individual capsules are learning, we take the average over capsule activations for each spatial location resulting in a vector of dimension C (C=number of capsules). Each feature in that C-dimensional vector shows the average activation (presence probability) of an individual capsule for that image. The highest activated capsule is used to sort the images into C groups. Figure 5 shows the images where a given capsule had the highest activation. In the figure, we can see different capsules are focused on different types of images, e.g., capsule 1 is mostly focused on *outdoor sports* like *surfing*. Since these capsule representations are learned in a weakly-supervised

(a) capsule 1: surfing, outdoor sports

(b) capsule2: food

(c) capsule 4: toys, zebra, horse

(d) capsule 5: elephants, wild animals

(e) capsule 6: sports, tennis

(f) capsule 9: pizza

(g) capsule 11: humans

(h) capsule 13: buildings

(i) capsule 15: bathroom, trains

Fig. 5: Images represented by individual capsules. Here, we show the group of images where a certain capsule has the highest activation, e.g., capsule 9 has the highest activation when there is *pizza* in the image.

manner, they show overlapping behavior over certain image classes. Some of them exhibit an interesting behavior. For instance, while capsule 2 is focused on *food items*, capsule 9 is fond of *pizza*; capsule 5 has learned what an *elephant* looks like, but also good at identifying *giraffes* and *cows* in the wild; capsule 13 is focused on *buildings*. We used our best model with C=16 capsules for these visualizations.

9 No. of training parameters

We compare the proposed model with other transformer-based methods in table 4. Our proposed system is shallower than the baseline methods using 5 layers in each modality-specific encoders followed by 2 cross-attentional layers. We denote the length of a vertical stack of transformer layers as the model's *depth*. We follow [9, 11] and consider one single modality layer as 1/2 of a multimodal layer. Hence, the proposed model has the depth=7 compared to the baselines with

Method	depth (#transformer layers)	#Params (M)
LXMERT [11]	12	239.8
ALBEF [10]	12	209.5
ViLT [9]	12	87.4
Ours	7	141.0

Table 4: Number of parameters in all transformer-based methods.

Answer Plausibility	ALBEF	ViLT	Ours
for all	92.12	92.28	**92.30**
for mispredicted	85.14	86.35	**87.15**

Table 5: Plausibility comparison on GQA-val set with ALBEF and ViLT for all question-answer pairs and the mispredicted question-answer pairs. We perform on par (even slightly better) than the baselines in terms of the predicted answer's plausibility. This verifies the system is predicting reasonable answers in the real-world context.

depth=12. In comparison to other transformer-methods, the proposed system uses less parameters ($\approx$ 141M) than LXMERT [11] (239.8M) and ALBEF [10] (209.5M). ViLT has the least number of parameters (87.4M). However, it is a single stream model compared to all other two-stream methods considered for this work including the proposed architecture. We excluded the text embedding layer when computing the number of training parameters since it is shared among all vision-language transformers which are used in this study [9].

10 VQA accuracy of ours vs. baselines:

Our proposed system while achieving better VQA accuracy than previous grounding SOTA on GQA dataset (MAC-Caps [8]]) and LXMERT (a transformer-based model with object-detection), performs lower than ViLT and ALBEF. We attribute this to two reasons: 1) Less training data – ViLT and ALBEF are using SBU and GCC additionally with strong data augmentations, so we assume that using additional data and comparable resources for training would improve our accuracy as well. 2) Considering failure cases in more detail, we find that the lower accuracy is mainly driven by semantically correct, but literally wrong answers such as girl vs women or herd vs cow (see examples in Fig. 6, 9, and 8). The answer prediction despite being reasonable is incorrect in terms of language mismatch with the ground truth. It could be possible that capsules help to prevent dataset biases, as they regularize and constrain the training and therefore suppress "shortcuts" based on dataset noise. To validate this further, we compute the plausibility metric for all questions as well as incorrect predictions. **Plausiblity** measures that an answer is reasonable in the real-world context e.g., it is unlikely to see a 'blue' apple in real-world. We perform on par with

ViLT and ALBEF for all predicted answers. When compared on the mispredicted questions for all three methods, our system predicts 2% more plausible answers than ALBEF and 0.8% better than ViLT (see table 5). This study maintains the observation about the predicted answer while being reasonable in the real-world is considered incorrect in terms of exact match with the ground truth consequently leading to decreased VQA accuracy.

11 Additional details about evaluation on VQA-HAT dataset

VQA-HAT dataset provides human attention maps for VQA task. This dataset is based on VQA v1.0 dataset and provides 1473 QA pairs with 488 images in validation set. To evaluate on this dataset, we train our system on VQA v1.0 and evaluate on VQA-HAT validation set. The answer vocabulary of VQA train set has a long tail distribution. We follow previous works [1, 3] and use 1000 most frequent answers. We first combine training (248,349 QA pairs) and validation data (121,512 QA pairs) to get a total of 368487 QA pairs. We then filter out the questions with out-of-vocabulary answers (answer vocab size is kept 1K) resulting in 318827 QA pairs. We separate out 10K QA pairs from the training set (after above mentioned question filtering) and use it as a validation set to pick our best model. We therefore use 308K QA pairs from VQA v1.0 train and val set for finetuning our pretrained backbone with 16 capsules. The learning parameters used for this training are lr=4e-5, batch size=64, with bert optimizer and trained for 20 epochs. The best model on validation set is used for evaluation.

12 Qualitative Results

In figure 6 and 7, we show more examples for qualitative comparison with baselines. Our system consistently produces correct grounding attention when compared to the baselines. In figure 8, we show some failure cases for our system in terms of grounding output as well as answer prediction. For grounding failure (in terms of IOU with the groundtruth box), we observe that the system's attention is cogent. For instance, in the top left example, for the question *where is the giraffe?*, the system is looking at the surroundings of giraffe and predicting the answer *zoo*. In the right two examples, the system is grounding correctly even generating reasonable answers. However, these answers are considered incorrect in terms of language mismatch with the groundtruth answer (herd vs. cow and beach vs. sand). Finally, in figure 9, we present more qualitative examples from our system.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)

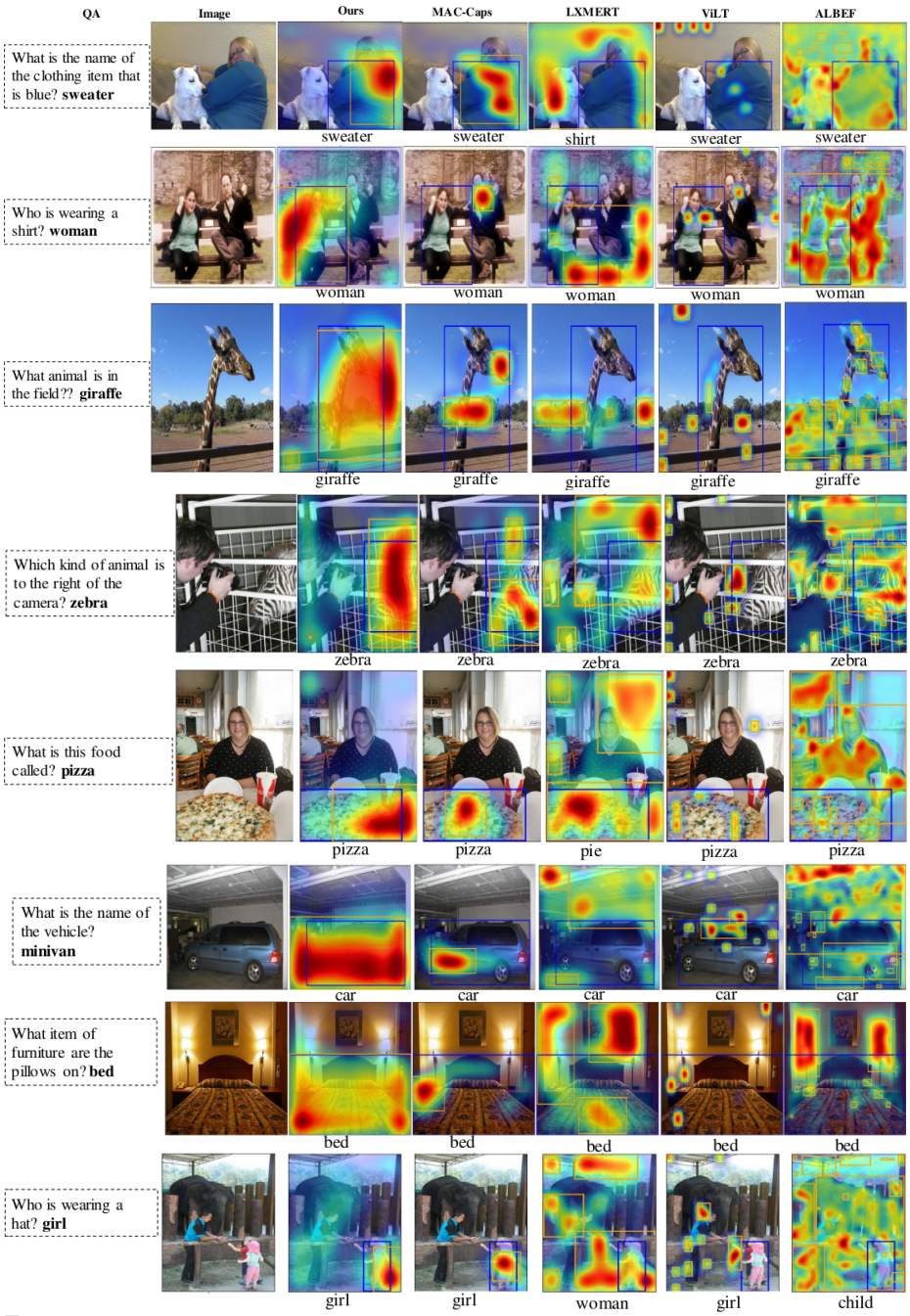


Fig. 6: More qualitative examples where the model predicted the answer correctly with attention (with detected orange boxes) on the correct image regions (blue boxes). Best viewed in color.

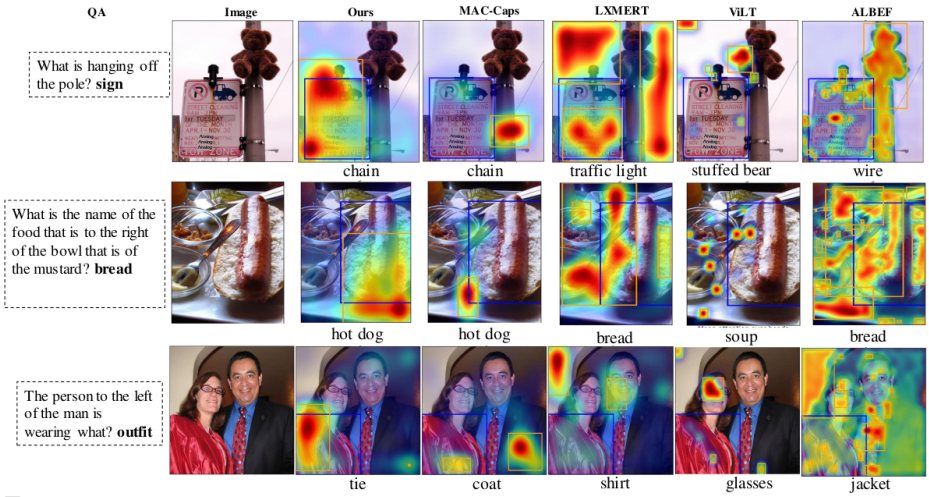


Fig. 7: Qualitative comparison for examples where our model predicted the wrong answer. The attention is over the correct image region.

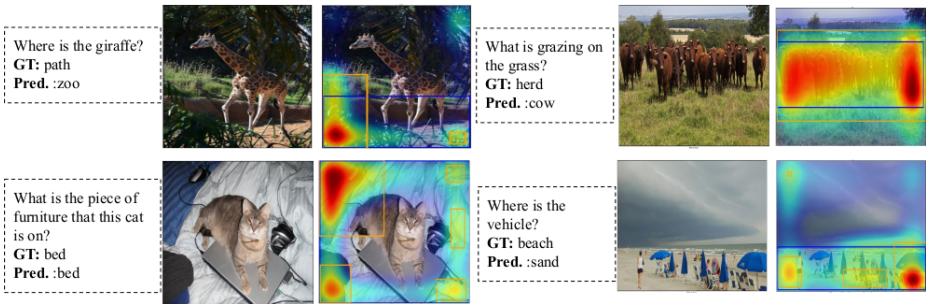


Fig. 8: Some failure cases for our system. Left two examples show the failure of grounding, right two examples are failure cases in terms of answer prediction. However, both the grounding output and the answers are plausible.

2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers (2021)
3. Das, A., Agrawal, H., Zitnick, C.L., Parikh, D., Batra, D.: Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)

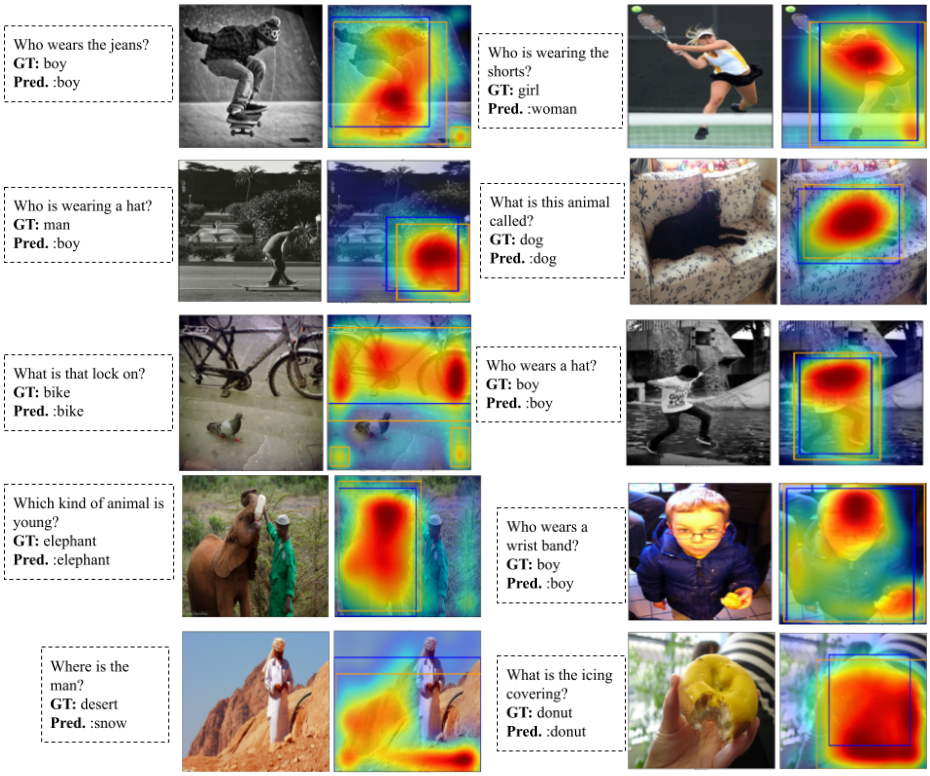


Fig. 9: More qualitative examples from our system.

6. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. International Conference on Learning Representations (ICLR) (2018)
7. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
8. Khan, A.U., Kuehne, H., Duarte, K., Gan, C., Lobo, N., Shah, M.: Found a reason for me? weakly-supervised grounded visual question answering using capsules (2021)
9. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 5583–5594. PMLR (18–24 Jul 2021), http://proceedings.mlr.press/v139/kim21k.html
10. Li, J., Selvaraju, R.R., Gotmare, A.D., Joty, S., Xiong, C., Hoi, S.: Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS (2021)
11. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (2019)

12. Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7w: Grounded question answering in images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4995–5004 (2016)