MDETR - Modulated Detection for End-to-End Multi-Modal Understanding Supplementary

Contents

A. Model details and hyperparameters

Pre-training hyperparameters MDETR follows the pre-train then fine-tune strategy by first training on our constructed combined dataset for 40 epochs followed by fine-tuning on the respective downstream task. We train our model using AdamW [12], a variant of Adam [9] better suited for weight decay. We use exponential moving average (EMA) with a decay rate of 0.9998, and a weight-decay of $1e^{-4}$. The backbone and the transformer have a constant learning rate of respectively $1e^{-4}$ and $1e^{-5}$ for 35 epochs, after which their learning rate is reduced by a factor of 10. For the language model's learning rate, we use a linear decay with warmup schedule, increasing linearly to $5e^{-5}$ during the first 1% of the total number of steps, then decreasing linearly back to 0 for the rest of the training.

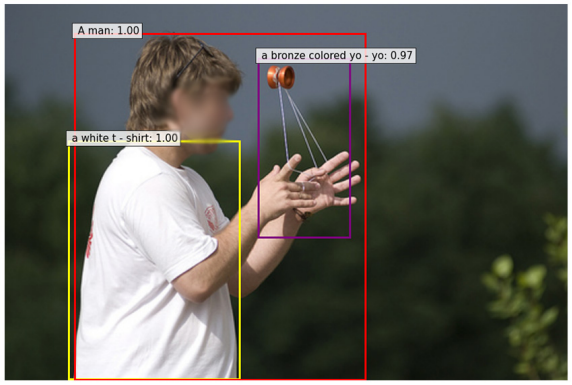
Figure 1: Illustration of the soft-token classification loss. For each object, the model predicts a distribution over the token positions in the input sequence. The weight of the distribution should be equally spread over all the tokens that refer to the predicted box.

Flickr30k Our results on Flickr30k are evaluated using the pre-trained model, without any additional fine-tuning as we found that it brings no additional gains. For evaluation, we must rank the boxes associated with each phrase. Since there might be several phrase in the same sentence, we must provide a ranking for each and every such phrase. To that end, we use the prediction from the soft-token classification loss. In the example depicted in Fig 1: to rank the boxes for the phrase "a cat", we use the probability mass that each query assigns to the positions that correspond to "a cat" in the sentence "a cat with white paws jumps over a fence in front of a yellow tree" (in this example, the first few tokens). Through this approach, the red box is found to be the highest-ranked box. On the other hand, if we want the boxes corresponding to "the fence", we sort them according to the corresponding token positions, and in this case we find the green box as the top-scoring one.

Referring Expression Comprehension For the Referring Expression Comprehension task, there is a stark difference in how the data is presented to the model in terms of density of annotation. For all other datasets that we use in our pre-training, each noun phrase in the sentence is annotated with its respective box, if available. On the other hand, in RE comprehension, the task is to align the whole referring expression with the corresponding box, possibly by needing to disambiguate between different occurrences of the same category of object. In other words, our model now needs to predict one box per expression. A problem with this setting is that our box-token contrastive alignment as well as soft token prediction losses get very diluted signal if we align the whole sentence to the box. To alleviate this, we pre-process the text using Spacy [4] to extract the root of the sentence using a dependency parser. The tokens from this root phrase are used to align to the box. Fine-tuning on this dataset is therefore crucial for good performance. We fine-tune for 5 epochs on the RefCOCO, RefCOCO+ and RefCOCOg datasets with a learning rate of $1e^{-5}$ for the backbone and $5e^{-5}$ for the rest of the network. We use a learning rate drop by a factor of 10 after 3 epochs. For the text encoder we use a learning rate of $1e^{-5}$, with a linear decay with warmup schedule, warming up over the first 1% of steps and then decaying to 0 linearly. At inference time, to detect a given expression, we feed it to the model alongside the image. We then rank the 100 detected boxes according to the probability that the box corresponds to an actual object (as opposed to a "no object"). If $P(\varnothing)$ is the probability mass assigned to the "no object" label, then we rank by increasing order of $P(\varnothing)$, or equivalently by decreasing order of $1 - P(\varnothing)$. We show an example in Fig 5 of the box predicted by our model for the corresponding referring expressions. In addition there is quite some variety in the type of text annotations from the three datasets. Both RefCOCO and RefCOCO+ were collected in a timed game setting whereas RefCOCOg was not. This led to differences in the length and diversity of language used in the different datasets. RefCOCO+ disallowed usage of location words to describe objects or disambiguate between multiple occurrences

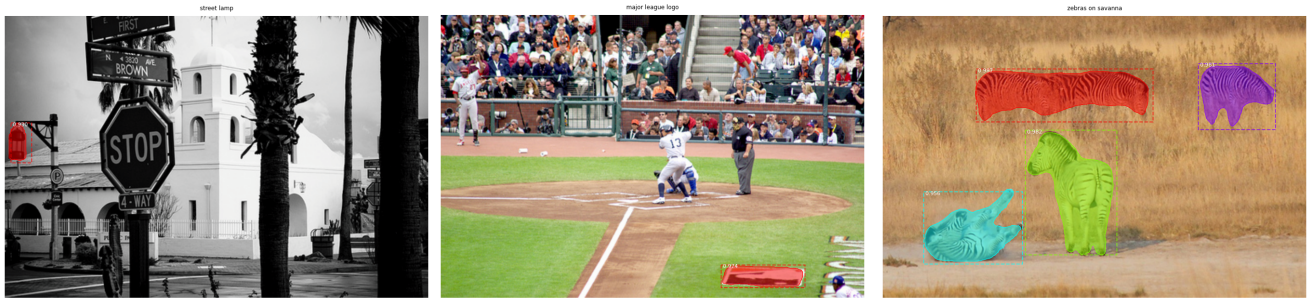
(a) "one small boy climbing a pole with the help of another boy on the ground"

(b) "A man talking on his cellphone next to a jewelry store"



(c) "A man in a white t-shirt does a trick with a bronze colored yo-yo"

Figure 2: Examples of phrase grounding on the Flickr30k dataset



(a) Query: "street lamp"

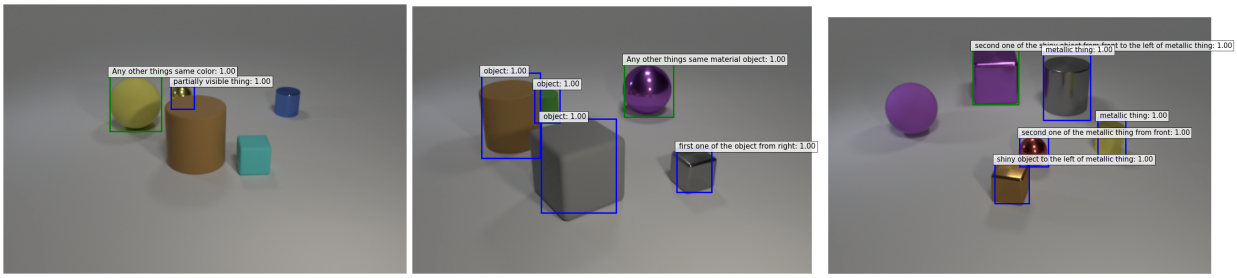
(b) Query: "major league logo"

(c) Query: "zebras on savanna"

Figure 3: Qualitative segmentation examples on the phrasecut dataset

of the same object, focusing more on appearance based descriptions. RefCOCOg consists of expressions more than twice the length (on average) of the others and with more flowery and descriptive language.

We also fine-tuned the EfficientNetb5 model on these datasets but did not see much improvement over the EfficientNetB3 model, and we believe this is due to the smaller size of these datasets causing the larger model to overfit.

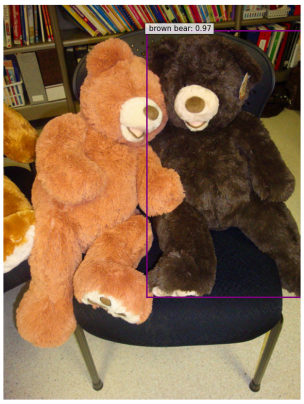


(a) Query: "Any other things that are the same color as the partially visible thing(s)"

(b) Query: "Any other things that are the same material as the first one of the object(s) from right"

(c) Query: "The second one of the shiny object(s) from front that are to the left of the second one of the metallic thing(s) from front"

Figure 4: Qualitative example from the CLEVR-REF+ dataset. When the model predicts a box that is referred to, we display it in green. The other boxes are intermediate reasoning steps and are depicted in blue.



(a) "brown bear"

(b) "zebra facing away"

(c) "the man in the red shirt carrying baseball bats"

(d) "the front most cow to the right of the other cows"

Figure 5: Examples from RefCOCO, RefCOCO+ and RefCOCOg datasets. Fig(a) taken from RefCOCO, Fig(b) from RefCOCO+ and Fig(c) and (d) are taken from RefCOCOg, in which the expressions are much longer on average and contain more descriptive language than in RefCOCO and RefCOCO+. Even when the expressions are long, we train our model to align the box to the root of the phrase, for eg. "the man" in (c). The model however, still has access to the whole text and uses it to disambiguate between the two men in the image.

PhraseCut Detection: For this phase, we use a batch size of 64, a learning rate of $1e^{-5}$ for the text encoder and backbone and $5e^{-5}$ for the rest of the network, and exponential moving average (EMA) of the network weights with a decay of 0.9998. segmentation: For this stage, we use a lr of $5e^{-4}$ and no EMA. See [1] for additional details.

Method	CLEVR						CLEVR-Humans		CoGenT		CLEVR-Ref+
	Overall	Count	Exist	Comp. Num	Query	Comp. Att	Before FT	After FT	TestA	TestB	Acc
MAttNet[24]	-	-	-	-	-	-	-	-	-	-	60.9
MGA-Net[27]	-	-	-	-	-	-	-	-	-	-	80.1
FiLM[15]	97.7	94.3	99.1	96.8	99.1	99.1	56.6	75.9	98.3	**78.8**	-
MAC [5]	98.9	97.1	99.5	99.1	99.5	99.5	57.4	81.5	-	-	-
NS-VQA[23]*	**99.8**	**99.7**	**99.9**	**99.8**	99.8	99.8	-	67.8	**99.8**	63.9	-
OCCAM [20]	99.4	98.1	99.8	99.0	99.9	99.9	-	-	-	-	-
MDETR	99.7	99.3	**99.9**	99.4	**99.9**	**99.9**	**59.9**	**81.7**	**99.8**	76.7	**100**

Table 1: Results on CLEVR-based datasets. We report accuracies on the test set of CLEVR, including the detail by question type. On CLEVR-Humans, we report accuracy on the test set before and after fine-tuning. On CoGenT, we report performance when the model is trained in condition A, without finetuning on condition B. On CLEVR-Ref+, we report the accuracy on the subset where the referred object is unique. *indicates method uses external program annotations

B. CLEVR Experiments

B.1. Dataset details

The CLEVR dataset consists of 3D-rendered scenes containing between 3 and 10 objects of various shapes, material, size and color. Each of these scenes is associated with about 10 questions that are formulated about the visible objects, generated from a fixed set of templates. Each question is guaranteed to be answerable, and the annotations further provide a functional program that describe how to compute the answer using elementary reasoning steps. The total training set contains 70k images and slightly less than 700k questions. Overall, the visual aspect of this task, *ie* the scene parsing, is not really challenging by modern standards, since the set of objects is limited, unambiguous, and there are no visual distractors. The only challenging cases occur in the event of heavy occlusion, where it might be hard to make out the shape of the occluded object, or in some cases where the question requires comparing ambiguous spatial relations (eg. asking which object is the closest to the camera in a setting where they are visually nearly tied). On the other hand, the text understanding aspect is more involved, since the questions can be quite complex, involving up to 20 reasoning steps. Unlike several successful approaches to CLEVR [7, 5, 14, 23] , MDETR doesn't incorporate any special inductive bias to cope with such complex reasoning tasks. In this section, we show that despite its relatively straight-forward formulation, our approach competes with state-of-the-art models on the question answering task.

The first ingredient required for training MDETR is bounding box annotations for objects in the image. The original CLEVR dataset doesn't provide any, so we use the scene graphs from the dataset to re-create the original scene in the 3D-renderer Blender, then use some of its functionalities to extract the segmentation masks of the visible parts of the objects, and deduce the bounding boxes from that. The main complication is that the original rendering involved some non-deterministic jittering of the camera's position and rotation, leading to some potential discrepancies in the computed boxes. To minimize the error, we use the known 3D position of each object, as well as their known 2D location in the rendered image to optimize the camera parameters using a gradient-based approach. The final boxes obtained using this approach are accurate within a 10-pixel error margin, which we deem appropriate for our purposes.

The second ingredient required is the alignment between bounding boxes and tokens in the question. MDETR is trained to predict *only* objects that are referred to in the question. For example, in the question "What is the color of the cube in front of the small cylinder?", we provide an annotation for both the small cylinder (an intermediate step) and the cube (the main subject), and none of the other objects present in the scene. We use the functional programs that are part of the original CLEVR annotations to extract this set of objects, along with their corresponding text tokens in the original question.

B.2. Training details

Model We use a ResNet-18 [3] from Torchvision, pre-trained on ImageNet [18] as the convolutional backbone. For the text-encoder, we use a pre-trained DistilRoberta [19] from HuggingFace [21]. The final transformer is the same as DETR, with 6 encoder layers as well as 6 decoder layers, and 8 attention heads in the attention layers. We reduce the number of object queries to 25, since the maximum number of objects to be detected is low.

Pre-training We first train the model only on the modulated detection objective, on our CLEVR-Medium subset, for 30 epochs. Following DETR training procedure, the transformer and the backbone use a learning rate of respectively $1e^{-4}$ and

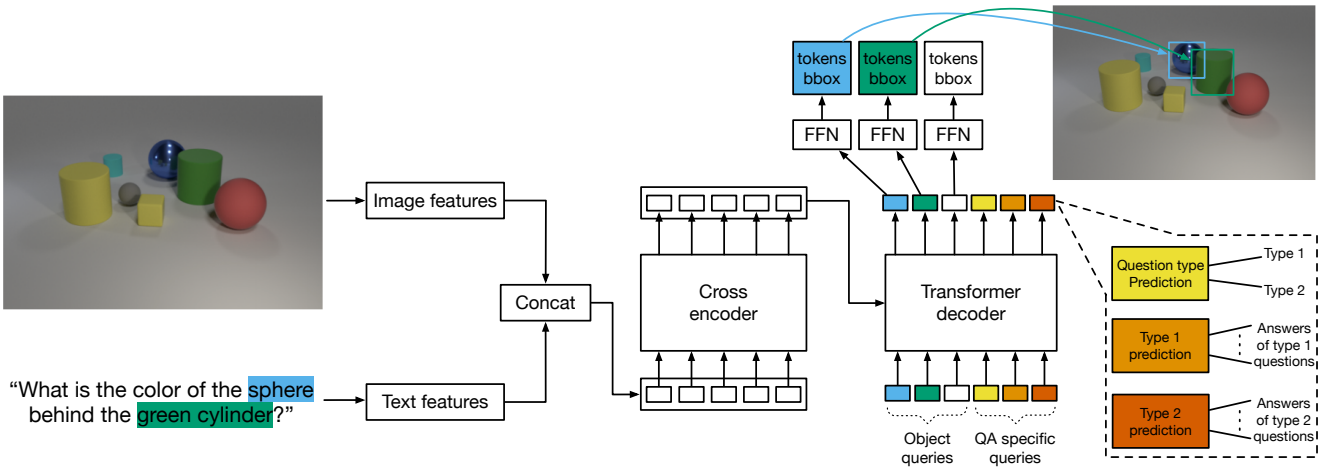


Figure 6: During MDETR pre-training, the model is trained to detect all objects mentioned in the question. To extend it for question answering, we provide QA specific queries in addition to the object queries as input to the transformer decoder. We use specialized heads for different question types.

$1e^{-5}$, and we reduce them by a factor of 10 at the 20th epoch. The text encoder uses a linear decay with warmup schedule, with a warm-up to $5e^{-5}$ over the first 1% of the training steps.

QA-finetuning For question answering, we take our pre-trained checkpoint, add the (untrained) question queries and their corresponding heads, then train on the full CLEVR dataset for 30 epochs, following the exact same learning rate schedule, with both the modulated detection as well as question answering losses. As depicted in Fig.12, we use additional queries in the transformer decoder to answer each type of question in CLEVR: numerical, binary and attributes. We supervise each of these heads using a standard cross-entropy loss. We monitor the accuracy on the validation set to apply early-stopping. Finally, for CLEVR-Humans, we further fine-tune for 60 epochs, with the learning-rate drop occuring at epoch 40.

B.3. Results and discussion

The results are collected in Table 1. On CLEVR, we closely match the performance of NS-VQA[23], a method that uses external supervision in the form of ground-truth program annotations, and clearly surpass the performance of methods which like us, don't use this extra supervision signal. We then evaluate the generalization capability of our model.

CLEVR-Humans [7] is a dataset of human-generated questions on CLEVR images. It tests the robustness of the model to new vocabulary and and different reasoning primitives. In a zero-shot setting, we improve substantially over the best competing model. We credit this improvement to our pre-trained language model. After fine-tuning, the gap narrows, suggesting that additional developments may be required to further improve performance.

CoGenT is a test for compositional generalization. The evaluation protocol consists in training on a set A, where the spheres can be any color but the cubes are either gray, blue, brown or yellow, and the cylinders are red, green, purple or cyan. We then evaluate in a zero-shot manner on a split B which has the opposite color-shape pairs for cubes and cylinders. Similar to other models, we observe a significant generalization gap. On closer inspection, the biggest drop in accuracy occurs on questions querying the shape of an object (from 99.98% on testA to 34.68% on testB), suggesting that the model has learnt strong spurious biases between shape and color.

CLEVR-REF+ Finally, we evaluate our model on CLEVR-REF+[11], a referring expression comprehension dataset built on CLEVR images. For each object query, we train an additional binary head to predict whether or not the query corresponds to an object being referred to (as opposed to an auxiliary object in the sentence, that we detect as well). Following [11], we evaluate accuracy on the subset of expressions that refer to a unique object, measured as whether the top ranked box has an IoU of at least 0.5 with the target box. Using the aforementioned binary prediction to rank the boxes, our model correctly ranks in first position a valid box for each of the examples of the validation set, leading to an accuracy of 100%, greatly outperforming prior work.

Model	AP
Baseline	99.0
- contrastive loss	83.2
- soft-token classification	87.7

Table 2: Ablation results on modulated detection on CLEVR-Medium. We report the class-agnostic AP. See text for details.

B.4. Ablations

We use CLEVR as a test bed to ablate several aspects of our model. Depending on the ablation, we report either the accuracy on the question answering task on the validation set of CLEVR, and/or the detection performance on this dataset, measured as a class-agnostic Average Precision (AP). When inspecting the modulated detection capabilities of the model, we use class agnostic Average Precision (AP) to evaluate the model. In the unconditional detection case, DETR is able to detect all boxes perfectly. When evaluated on the task of modulated detection, the AP metric therefore captures the model's capability for text understanding since now only the boxes relevant to the query must be detected. To put this in context, a model that detects all boxes even when given a text query (thereby ignoring the text completely) gets an AP of around 60. The goal is to achieve an AP close to 100 which would imply the model only finds the relevant boxes.

B.4.1 Loss ablations

We first ablate the various parts of our loss. The results are summarized in Table 2 We report modulated detection results on the CLEVR-Medium subset, that we constructed by removing data-points from CLEVR where the same object is referred to by distinct parts of the question. In these ablations, we consider only the performance of the detector, and not the question answering capability. As a result, the queries related to question answering are not present and we do not propagate any QA related loss.

Contrastive loss We first ablate the impact of the contrastive loss, by training a model without it. In this situation, the alignment must occur solely through the soft-token classification loss. As shown in Table 2, removing this loss results in a drastic drop in AP. More specifically, when evaluating the model, it becomes apparent that it is able to filter the objects based on some attributes (in particular their shape and size) but not others (in particular color and texture). It is unclear what drives the model in this local sub-optima, nor what statistical shortcut it is leveraging to correctly identify shapes and sizes. However, it shows that solely predicting the spans of the text query associated with each object is not sufficient to learn proper alignment. The contrastive loss, which forces object-queries to be similar to their corresponding text-token, is thus necessary.

Soft-token classification loss We now study whether predicting the text-spans associated with each object is necessary, provided that we propagate the contrastive loss. Instead of predicting a distribution over span, we construct a simplified version of MDETR which only predicts a binary label for each object query: "object" or "no-object" ($\varnothing$). This formulation is equivalent to the vanilla DETR classification loss, with one object class. We observe similar results as the previous ablation, namely a sharp decline in AP and a model that only understands half of the attributes correctly. We thus conclude that both ingredients of our loss are indeed required.

B.4.2 Question answering ablations

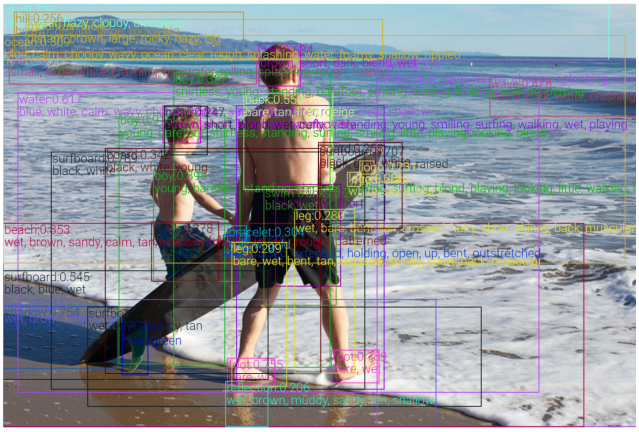
Finally in Table 3 we ablate two aspects of our training recipe that differ with previous approaches:

- **Curriculum**: We evaluate a model trained directly on the full CLEVR training set, without our modulated detection pretraining on CLEVR-medium. Similarly as the previous section, the model learns to detect only a subset of attributes, leading to poor QA accuracy.

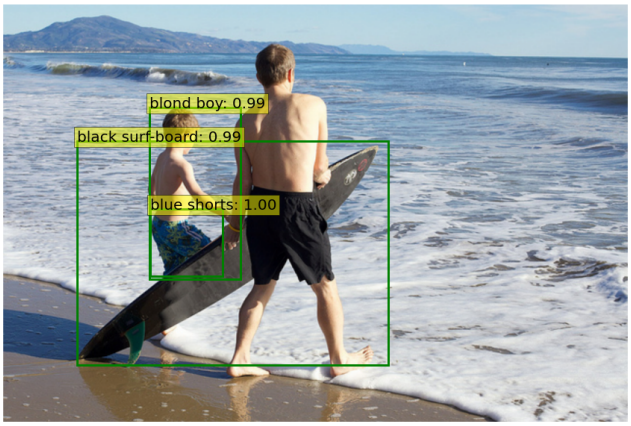
- **Single QA head** In our approach, we train a specialized head for each type of question (numerical, binary, or categorical over the attributes). This differs from previous approaches that usually cast it as a single classification over all possible answers. As shown in Table 3, this separation has a big impact on the final accuracy. We hypothesize that it enables the attention pattern for each question type to specialize accordingly to the task, there-by yielding better performance.

Model	Detection AP	QA accuracy
Baseline	99.0	99.7
No curriculum	89.7	68.2
Single QA head	99.0	90.1

Table 3: Ablation results on the validation set of CLEVR. We report the class-agnostic AP and the question answering accuracy. See text for details.



(a) Current object detection pipeline outputs, predicting all possible objects in the image. This extensive annotation is essential to multi-modal understanding systems that treat detection as a black box.



(b) MDETR predicts boxes relevant to the caption and labels them with the corresponding spans from the text. Here we use the caption: "blond boy wearing blue shorts. a black surf-board."

Figure 7: Modulated detection using MDETR vs detection output for a current state-of-the-art multi-modal understanding system. Image taken from [26]

C. Dataset constructions

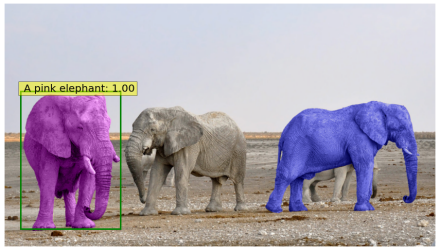
MS COCO On the COCO dataset, we include annotations from the referring expressions datasets (RefCOCO [25], RefCOCO+ [25] and RefCOCOg [13] datasets). By construction, in this dataset, each referring expression is a whole sentence that describes one object in the image, where the constituent noun phrases from the sentences are not themselves annotated. For example, in Figure **??**, the caption would be "the person in the grey shirt with a watch on their wrist.", where only the person would be annotated and not the grey shirt or their watch. To avoid ambiguity, we perform some text pre-processing using SpaCy [4] to extract the *root* of the referring expression. This is used in our soft token prediction as well as the contrastive alignment loss for aligning to the referred box. The auxiliary objects (in this example the shirt and the watch) are ignored altogether.

Visual-Genome We use annotations from VG regions, a dataset having diverse descriptions of a wide variety of objects, often having a very high degree of descriptive detail and covering several concepts. By construction, the VG dataset comprises a lot of redundant annotations. We detect redundant sentences by normalizing them (removing all punctuation, stop-words, and lower-casing), then testing for equality. Once we found a pair of equivalent sentences, two cases arise:

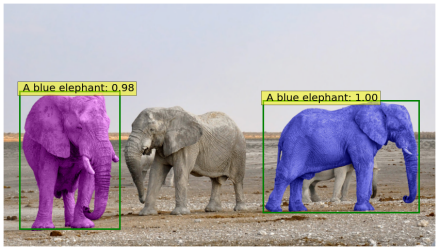
- The corresponding boxes are highly overlapping (IoU > 0.7). In this case, we consider both annotations to be redundant, and we keep only one of them.
- The boxes are non-overlapping. The most likely explanation is that the sentence is under-specified and actually corresponds to several distinct objects in the image. In this case, we merge the two data-points together, and the resulting annotations comports two boxes for this sentence.

We iterate recursively this process until no equivalent sentences remain.

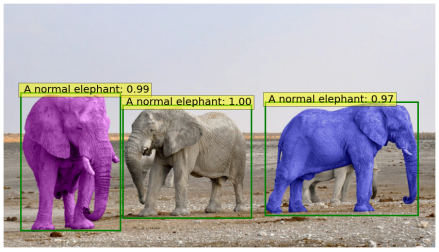
In some cases, the VG annotations provide information about the object referred to in the sentence. For example, if the region is tagged "the cat on the white table", in some cases the individual boxes for the cat and the table are available. In



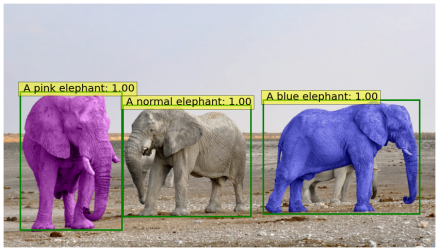
(a) Text prompt: "A pink elephant."



(b) Text prompt: "A blue elephant."



(c) Text prompt: "A normal elephant."



(d) Text prompt: "A pink elephant. A blue elephant. A normal elephant"

Figure 8: Qualitative results on unseen attributes combinations. While the model correctly singles out the pink elephant (a), it incorrectly includes the pink elephant when prompted about the blue one (b). In (c), we show that the model does not understand what a "normal elephant" looks like. However, in (d), when prompted about all three elephants at once, it is able to assign the correct label to each of them, by process of elimination.

this case, we discard the region box and use the individual boxes instead. We note that despite our merging strategy, it may remain some region description that do not canonically refer to a unique object in the image, but for which we don't have ground-truth annotations for the other objects that also match the said region description. Despite the noise it introduces in the training process, we don't pursue extra efforts to try and fix these situations. In addition, we also use questions from the GQA dataset [6], where bounding box annotations are provided for key phrases in the questions.

D. Evaluating grounded detection

The main evaluation metric proposed to evaluate grounded detection in datasets like Flickr30K entities[17] is Recall@k, that is measuring whether a given model is able to rank the "correct" box amongst the top k it produces. The correctness of a box is decided by computing the Intersection-over-Union (IoU) between the proposed box and the ground-truth box, and deemed correct if the IoU is above a predetermined threshold, generally 0.5. While this kind of evaluation is well-suited for tasks where there is a clear one-to-one mapping between phrases and boxes, for example in Referring Expression Comprehension tasks, we argue that in general grounded detection tasks, they fall short of properly evaluating the performance of the models. Specifically, they run into the following issues:

1. **Multiple boxes for a given phrase**: Since the recall@k metric implies a single box per phrase, it is not clear how to extend it to situations where a given phrase refers to several distinct objects in the image . In the absence of clear guidelines, various authors have adopted divergent approaches to deal with that. Specifically, some [10, 8] consider the predicted box to be correct if it has an IoU>0.5 with *any* of the ground-truth boxes. We refer to this protocol as the ANY-Protocol. Others [16, 22] first merge all the ground-truth boxes associated to the phrase by considering the smallest enclosing box. Then they proceed to compute the IoU as usual, using this merged box as the ground-truth one. We refer to this protocol as the MERGED-BOXES-Protocol.

   Arguably, both methods have drawbacks: the first one keeps the atomicity of each instance but fails to evaluate whether the model found all the referred instances. The second one looses the fine-grained instance in favor of a box that may be unreasonably bloated if the instances are spread apart.

2. **Multiple phrases for a given box**. In some cases, the same object is referred to multiple times in the sentence. Sometimes, the corresponding phrases are exact duplicates of each other, or close synonyms (eg "a guy" and "a man"), but

sometimes the co-references are more subtle. One such example include referring alternatively to a group (eg "a couple") or to a sub-constituent (eg "the woman").

Under the current evaluation protocol, each phrase is evaluated independently, even if they refer to the same object. As such, it does not test the model's understanding of co-references, which is arguably an important aspect of learning grounded representations.

Recognising the discrepancy in the evaluation procedures in the literature and the difficulties it creates in comparing various approaches, we also evaluate MDETR under the merged-box protocol, as described in Sect D.1.

D.1. Evaluation under the MERGED-BOXES-Protocol

In MDETR, the dataset creation process operates under the assumption that a phrase is associated to all the individual boxes that correspond to it. As a result, MDETR does not naturally predict the merged-boxes that would be required by the MERGED-BOXES-Protocol. For that reason, it is necessary to fine-tune the model on a version of Flickr30k entities where the boxes have been merged appropriately. We note that the ANY-Protocol did not require such fine-tuning.

E. Error Analysis

To better understand the failures of the model on the grounding task, we provide a small-scale error analysis. We evaluate our best model, the EfficientNetB5 variant, on the validation set of Flickr30k. We manually inspect the first 100 errors made by the model. The results are summarized in Fig 9 and we detail the types of errors we uncovered in the following:

- Issues with the ground truth annotations

  Ambiguous GT location This corresponds to phrases that don't have a canonical localization. They mostly correspond to scene elements, such as "beach" or "tree".

  Inconsistent annotations (when several objects are referred) In the ANY-Protocol, if a phrase corresponds to several objects, then in principle every single instance should be individually annotated. However, we find some inconsistencies in the annotations, for example some instances that are missed, or some distinct instances that are annotated using the same box.

  Imprecise GT box This corresponds to cases where the provided box is not adequate. It is either too big (not tight), or too small, cutting out a part of the object.

  Wrong GT In those cases, the annotated box(es) don't correspond to the correct referred object at all.

- Grounding mistakes

  Wrong instance The model picks an instance of the correct type (including adjective modifiers, if any), however it is not the correct instance when taking into account context from the rest of the sentence.

  Wrong object The model picks the wrong object, and it is not of the correct type. This usually occurs on long-tail concepts that the model doesn't seem to know about.

  OCR The phrase refers to written text, thus requiring OCR abilities from the model, which it doesn't have.

- Detection issues

  Imprecise box prediction The model is clearly selecting the right object, however the predicted box isn't quite precise enough. This happens on small objects as well as elongated objects, where a relatively small L1 error can cause a low IoU with the ground truth.

Overall, we find that on the analyzed subset, more than half the errors stem from issues in the ground-truth annotations. Extrapolating to the rest of the dataset, this would imply a label noise of around 10%, which might be detrimental to make significant further progress on the task.

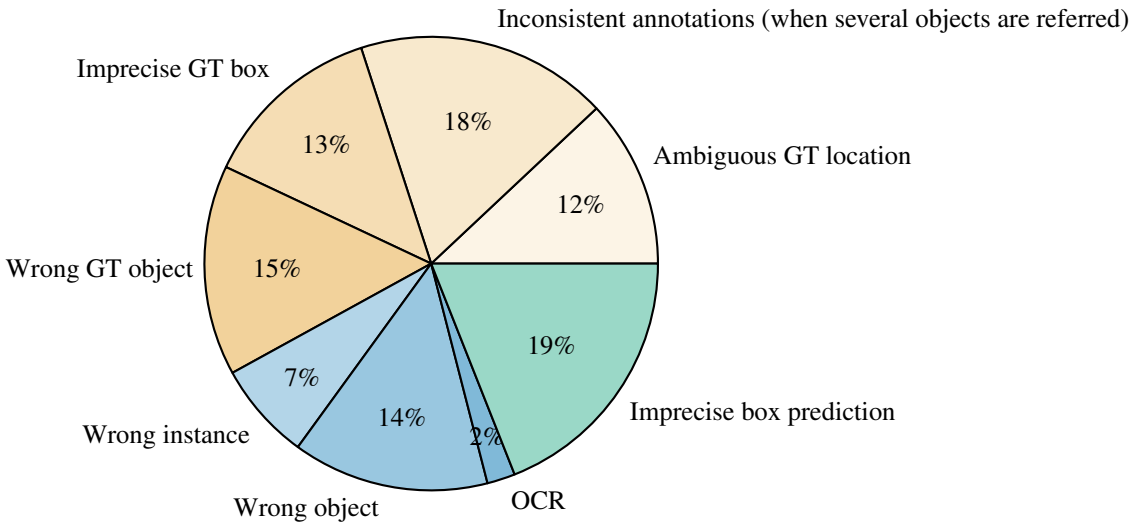


Figure 9: Break down of the first 100 errors from the EfficientNetB5 model on the Flickr30k validation set. Yellow shade corresponds to mistakes in the ground truth annotations. Blue shade corresponds to errors made by the model in grounding. Green shade corresponds to errors made by the model in accurate localization. See text for more details.

F. Experiments on VQA2

In our visual question answering experiments on the GQA dataset, we always had access to bounding box information for the questions. Only during fine-tuning on the balanced set for 10 epochs, we do not supervise the detection losses. On the other hand, for datasets such as VQA2 [2], we do not have access to any box annotations and the supervision comes solely from the question answering loss. We fine-tune two of our models — pre-trained on the joint dataset, as well as the model fine-tuned on the all-split of GQA — on the VQA v2 dataset for 25 epochs. The results are reported in Table 4. While these results are not state-of-the-art on the VQA2 benchmark, they are still quite reasonable with respect to current literature. This show that our method can be extended to tasks where we do not have the dense supervision (in the form of bounding boxes and their alignment to the text) that we otherwise expect in tasks reported in this paper.

Pre-training	Test-Dev				Test-Std			
	Overall	Yes-no	Other	Number	Overall	Yes-no	Other	Number
Modulated detection on combined dataset (40 epochs)	70.49	86.74	55.17	60.04	-	-	-	-
+ GQA-*all* (5 epochs)	70.64	86.74	55.26	60.33	70.63	86.79	55.12	60.12

Table 4: VQA v2 results

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 4

[2] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 11

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[4] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. 2, 8

[5] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. *ArXiv*, abs/1803.03067, 2018. 5

[6] Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. *arXiv:1902.09506 [cs]*, May 2019. arXiv: 1902.09506. 9

[7] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Inferring and Executing Programs for Visual Reasoning. *arXiv:1705.03633 [cs]*, May 2017. arXiv: 1705.03633. 5, 6

[8] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *arXiv preprint arXiv:1805.07932*, 2018. 9

[9] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2

[10] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 9

[11] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan Yuille. CLEVR-Ref+: Diagnosing Visual Reasoning with Referring Expressions. 6

[12] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. 2

[13] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous Object Descriptions. *arXiv:1511.02283 [cs]*, April 2016. arXiv: 1511.02283. 8

[14] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning. pages 4942–4950. 5

[15] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. 5

[16] Bryan Allen Plummer, Kevin Shih, Yichen Li, Ke Xu, Svetlana Lazebnik, Stan Sclaroff, and Kate Saenko. Revisiting image-language networks for open-ended phrase detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 9

[17] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 9

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015. 5

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. 5

[20] Zhonghao Wang, Mo Yu, Kai Wang, Jinjun Xiong, Wen-mei Hwu, Mark Hasegawa-Johnson, and Humphrey Shi. Interpretable Visual Reasoning via Induced Symbolic Space. 5

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020. 5

[22] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4683–4693, 2019. 9

[23] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. 5, 6

[24] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. MAttNet: Modular Attention Network for Referring Expression Comprehension. *arXiv:1801.08186 [cs]*, March 2018. arXiv: 1801.08186 version: 3. 5

[25] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling Context in Referring Expressions. *arXiv:1608.00272 [cs]*, August 2016. arXiv: 1608.00272 version: 3. 8

[26] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *arXiv preprint arXiv:2101.00529*, 2021. 8

[27] Yihan Zheng, Zhiquan Wen, Mingkui Tan, Runhao Zeng, Qi Chen, Yaowei Wang, and Qi Wu. Modular graph attention network for complex visual relational reasoning. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020. 5