



# TxT: Crossmodal End-to-End Learning with Transformers

Jan-Martin O. Steitz<sup>1</sup>(✉) , Jonas Pfeiffer<sup>1</sup> , Iryna Gurevych<sup>1,2</sup> ,  
and Stefan Roth<sup>1,2</sup> 

<sup>1</sup> Department of Computer Science, TU Darmstadt, Darmstadt, Germany  
jan-martin.steitz@visinf.tu-darmstadt.de

<sup>2</sup> hessian.AI, Darmstadt, Germany

**Abstract.** Reasoning over multiple modalities, *e.g.* in Visual Question Answering (VQA), requires an alignment of semantic concepts across domains. Despite the widespread success of end-to-end learning, today’s multimodal pipelines by and large leverage pre-extracted, fixed features from object detectors, typically Faster R-CNN, as representations of the visual world. The obvious downside is that the visual representation is not specifically tuned to the multimodal task at hand. At the same time, while transformer-based object detectors have gained popularity, they have not been employed in today’s multimodal pipelines. We address both shortcomings with *TxT*, a transformer-based crossmodal pipeline that enables fine-tuning both language and visual components on the downstream task in a fully end-to-end manner. We overcome existing limitations of transformer-based detectors for multimodal reasoning regarding the integration of global context and their scalability. Our transformer-based multimodal model achieves considerable gains from end-to-end learning for multimodal question answering.

**Keywords:** Vision and language · Transformers · End-to-end learning

## 1 Introduction

Vision and language tasks, such as Visual Question Answering (VQA) [2, 11, 44], are inherently multimodal and require aligning visual perception and textual semantics. Many recent language models are based on BERT [6], which is based on transformers [45] and pre-trained solely on text. The most common strategy to turn such a language model into a multimodal one is to modify it to reason additionally over image embeddings, which are typically obtained through some bottom-up attention mechanism [1] and pre-extracted. The consequence is that the visual features cannot be adjusted to the necessities of the multimodal reasoning task during training. Jiang *et al.* [17] recently showed that fine-tuning the visual representation together with the language model can benefit the accuracy in a VQA task. Pre-trained image features are

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-030-92659-5\\_26](https://doi.org/10.1007/978-3-030-92659-5_26).

thus not optimal for the multimodal setting, even when pre-training with semantically diverse datasets, e.g. Visual Genome [20]. In this work, we follow [17] to ask how the visual representation can be trained end-to-end within a multimodal reasoning pipeline. In contrast, however, we focus on recent transformer-based architectures [5, 6] with their specific requirements.

Multimodal pipelines typically [44] rely on visual features from the popular Faster R-CNN object detector [37]. As Faster R-CNN makes heavy use of subsampling to balance positive and negative examples during training and needs non-maximum suppression to select good predictions, it is not easily amenable to end-to-end training as desired here. Therefore, [17] remodels Faster R-CNN into a CNN that produces dense feature maps, which are combined with an LSTM-based model [48] and trained end-to-end. It is difficult though to combine such dense features with the latest transformer-based language models, e.g. [5, 22, 29, 43], as the large number of image features causes scalability issues in the underlying pairwise attention mechanism. We thus follow a different route, and instead propose to employ an alternate family of object detectors, specifically the recent Detection Transformers (DETR) [4] and variants, which treat the detection task as a set-prediction problem. DETR produces only a small set of object detections that do not need any treatment with non-maximum suppression as its transformer decoder allows for the interaction between the detections. We are thus able to employ the full object detector and only need to reason over a comparatively small set of image features in the BERT model. However, a number of technical hurdles have to be overcome.

Specifically, we make the following contributions: (i) We introduce alternative region features for image embeddings produced by DETR [4] and Deformable-DETR [49], two modern, transformer-based approaches to object detection. Out of the box and despite their competitive accuracy for pure object detection, these transformer-based detectors deteriorate the VQA accuracy compared to the common Faster R-CNN bottom-up attention mechanism [1]. (ii) We show that this accuracy loss stems from the global context of the detected objects not being sufficiently represented in the bottom-up features. We mitigate this effect through additional global context features, which bring the VQA accuracy much closer to the baseline. While transformer-based detectors, such as DETR, are powerful, they are also very computationally heavy to train; faster to train alternatives, *i.e.* Deformable-DETR [49], are less efficient at test time, which is also undesirable. (iii) We address this using a more scalable variant of Deformable-DETR that still leverages multi-scale information by querying the multi-scale deformable attention module with only one selected feature map instead of a full multi-scale self-attention. This allows retaining the training efficiency of [49], yet is comparably fast at test time as DETR. (iv) Our analysis shows that Deformable-DETR, while being a stronger object detector than DETR, also modifies the empirical distribution of the number of detected objects per image. We find that this negatively impacts the VQA accuracy and trace this effect to the use of the focal loss [24] within Deformable-DETR. (v) Our final transformer-based detection pipeline enables competitive VQA accuracy when combined with transformer-based language models despite having a shorter CNN backbone compared to [37]. More importantly, the full model including the visual features can be trained end-to-end, leading to accuracy gains of 2.3% over pre-extracted DETR features and 1.1% over Faster R-CNN features on the VQA<sub>v2</sub> dataset [11].

## 2 Related Work

With the advancement of transfer learning, the current modus operandi for multimodal learning is to combine pre-trained models of the respective modality by learning a shared multimodal space. We thus review approaches for single modalities before turning to crossmodal reasoning.

### 2.1 Learning to Analyze Single Modalities

*Language Understanding.* Recently, transfer learning has dominated the domain of natural language processing, where pre-trained models achieve state-of-the-art results on the majority of language tasks. These models are predominantly trained with self-supervised objectives on large unlabeled text corpora, and are subsequently fine-tuned on a downstream task [14, 31]. Most recent language models leverage the omnipresent transformer architectures [45] and are trained predominantly with Masked-Language-Modelling (MLM) objectives as *encoders* [6, 27], with next-word prediction objectives as *generative/decoder* models [21, 35], or as *sequence-to-sequence* models [3, 33, 34].

*Visual Scene Analysis.* Convolutional neural networks (*e.g.*, ResNet [12]) are widely used to encode raw image data. While originating in image classification tasks on datasets like ImageNet [38], they are much more broadly deployed with transfer learning as backbone for other tasks like object detection [10] or semantic segmentation [41].

The dominant methods for object detection are Faster R-CNN [37] and variants like Feature Pyramid Networks (FPN) [23] due to their high accuracy [15]. Faster R-CNN has a two stage architecture with a shared backbone: First, a region proposal network suggests regions by a class-agnostic distinction between foreground and background. After non-maximum suppression, the highest scoring regions are fed to the region-of-interest (RoI) pooling layer. RoI pooling extracts fixed-sized patches from the feature map of the shared backbone, spanning the respective regions. These are sent to the second stage for classification and bounding box regression. A final non-maximum suppression step is necessary to filter out overlapping predictions. Single-stage object detectors, *e.g.* SSD [26] and YOLO [36], offer an alternative, but still need sampling of positive and negative examples during training, hand-crafted parts for *e.g.* anchor generation, and a final non-maximum suppression step. More recently, Detection Transformers (DETR) [4] were proposed with a transformer-based architecture, which is conceptually much simpler and can be trained without any hand-crafted components. Deformable-DETR [49] introduced a multi-scale deformable attention module to address DETR’s long training time and its low accuracy for small objects.

### 2.2 Learning Multimodal Tasks

*Overview.* Most recent work on crossmodal learning relies on combining pre-trained single-modality models to learn a shared multimodal space. To do so, both image and text representations are passed into a transformer model [5, 22, 29, 43], where a multi-head attention mechanism reasons over the representations of both modalities. The

transformer model is initialized with the weights of a pre-trained language model. While word-embeddings represent the text input, raw images are passed through pre-trained vision models that generate encoded representations, which are passed into the transformer. While ResNet encodings of the entire image can be leveraged [19], it has been shown that utilizing object detection models (*i.e.* Faster R-CNN [37]), which provide encoded representations of multiple regions of interest [1], benefits the downstream task [29, 43, *inter alia*]. Here, the image features are passed through an affine-transformation layer, which learns to align the visual features with the pre-trained transformer. Similarly, the pixel offsets<sup>1</sup> are used to generate positional embeddings. By combining these two representations, each object region is passed into the transformer separately.<sup>2</sup>

*Datasets.* To learn a shared multimodal representation space, image captioning datasets such as COCO [25], Flickr30k [32], Conceptual Captions (CC) [40], and SBU [30] are commonly utilized. The self-supervised objectives are largely the same among all approaches: next to MLM on the text part, masked feature regression, masked object detection, masked attribute detection, and cross-modality matching.

*Transformer Approaches.* Recent multimodal models initialize the transformer parameters with BERT [6] weights and leverage the Faster R-CNN object detection model: LXMERT [43] and ViLBERT [29] propose a dual-stream architecture, which provides designated language and vision transformer weights. A joint multi-head attention component attends over both modalities at every layer. UNITER [5] and Oscar [22] propose a single-stream architecture, which shares all transformer weights among both modalities. Oscar additionally provides detected objects as input to the transformer, and argues that this allows for better multimodal grounding. VILLA [8] proposes to augment and perturb the embedding space for improved pre-training.

*End-to-End Training.* Above approaches combine pre-trained vision and language models, however, they do not back-propagate into the vision component. Thus, no capacity is given to the vision model to reason over the raw image data in terms of the downstream task; the assumption is that the pre-encoded representations are sufficient for the downstream crossmodal task. Kamath *et al.* [18] avoid this by incorporating multimodality into a transformer-based object detector. It is end-to-end trainable but needs computationally heavy pre-training. Jiang *et al.* [17] address this by proposing to extract the Faster R-CNN weights from [1] into a CNN. They are able to leverage multimodal end-to-end training, but use a model based on long short-term memory (LSTM) [13]. The sequential processing of LSTM models hinders parallelization and thus limits the sequence length. Pixel-BERT [16] proposes to embed the image with a CNN and reasons over all resulting features with a multimodal transformer. As such their model is end-to-end trainable, but very heavy on computational resources because of the pairwise attention over these dense features. For VL-BERT [42], a Faster R-CNN is used

<sup>1</sup> Depending on the model, this includes relative position, width, and height of the original image.

<sup>2</sup> Different approaches for combining pixel offsets with object features have been proposed, such as adding the representation (*e.g.*, used by LXMERT [43]) or transforming them jointly (*e.g.*, used by Oscar [22]).

to pre-extract object bounding boxes. The region proposal network of Faster R-CNN is then excluded in the further training process and the language model is jointly trained with the object detector in a Fast R-CNN [9] setting. This separates region proposals from object classification and localization for test time, hence multi-stage inference is necessary and no full end-to-end training is possible.

To mitigate this, we propose TxT, a transformer-based architecture that combines transformer-based object detectors with transformer-based language models and can be trained end-to-end for the crossmodal reasoning task at hand.

### 3 Transformers for Images and Text

Transformers [45] not only quickly developed into a standard architecture for language modeling [6], but their benefits have recently been demonstrated also in visual scene analysis [4, 7, 49]. We aim to bring these two streams of research together, which aside from conceptual advantages will allow for end-to-end learning of multimodal reasoning.

#### 3.1 Transformer-Based Object Detection

*DETR.* The Detection Transformer (DETR) [4] is a recent transformer-based object detector, which treats object detection as a set-prediction problem and, therefore, obviates the need for non-maximum suppression. It outperforms the very widely used Faster R-CNN [37] for object detection on standard datasets like COCO [25]. DETR first extracts a feature map from the input image using a standard CNN backbone. The features are then projected to a lower dimensionality through  $1 \times 1$  convolutions. A subsequent transformer encoder uses multi-head self-attention layers and pointwise feed-forward networks to encode the features. These are then passed to the decoder, where a fixed set of learned object queries is used to embed object information by cross-attention with the encoder output. Finally, feed-forward networks predict classes and bounding boxes. The predictions are connected by bi-partite matching with the ground truth and a set-based loss is used for training. As such, DETR is completely end-to-end trainable without hand-crafted parts.

*Deformable-DETR.* Deformable-DETR [49] addresses two important drawbacks of the original DETR architecture: The long required training time (500 epochs are needed for full training) and its relatively low detection accuracy for small objects. Deformable-DETR replaces the standard multi-head attention layers with multi-scale deformable attention modules in its transformer layers. Instead of a pairwise interaction between all queries and keys, they only attend to a small set of sampling points around each reference point, interpolated at learned position offsets. To incorporate multi-scale information, the attention modules work on a set of feature maps of different scales and each query feature attends across all scales. Additionally, in Deformable-DETR the focal loss [24] is used for matching between predictions and ground truth as well as scoring the classification. Deformable-DETR is not only able to reduce the total training time to only one  $10^{\text{th}}$  of that of DETR, but even outperforms it for object detection on standard datasets [25].

**Table 1.** Comparison of object detector architectures on the COCO validation set [25] in terms of average precision (AP, higher is better) using the COCO detection evaluation metrics. The results for DETR, DETR-DC5, and Faster R-CNN are quoted from [4] and the multi-scale Deformable-DETR evaluation is taken from [49]. The (+) in Faster R-CNN indicates that the models were trained with an additional GIoU loss. The GFLOPs are measured over the first 100 images of the COCO validation set. Deviating from [4,49], which estimate the number of floating point operations and omit part of the model, we log all CUDA calls with NVIDIA Nsight Compute, therefore recording higher – but more reliable – numbers.

Model	GFLOPs	#params	Epochs	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
Faster R-CNN-FPN+	253	41M	109	42.0	62.1	45.5	26.6	45.4	53.4
Faster R-CNN-R101-FPN+	360	60M	109	44.0	63.9	47.8	27.2	48.1	56.0
DETR	161	41M	500	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	490	41M	500	43.3	63.1	45.9	22.5	47.3	61.1
Deformable-DETR <i>query stride 16</i>	177	40M	50	42.9	62.9	45.9	24.2	46.1	60.4
Deformable-DETR <i>multi-scale</i>	337	41M	50	45.4	64.7	49.0	26.8	48.3	61.7

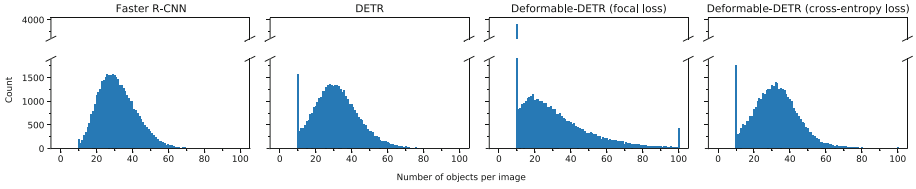
### 3.2 Global Context for DETR Models

Before we can bring together transformer architectures for object detection and language modeling in an end-to-end fashion, we first need to assess whether transformer-based detectors yield a competitive baseline. To that end, we use pre-extracted features from pre-trained detectors in visual question answering.

Surprisingly, we find that replacing Faster R-CNN features by those from DETR leads to a noticeable 2.5% accuracy loss in terms of VQA<sub>v2</sub> [11], even if DETR is a stronger detector on standard object detection benchmarks [25]. Looking at this more closely, we find that in contrast to Faster R-CNN, DETR and Deformable-DETR encode the semantic information in a single feature vector for each query in the transformer-based decoder. We posit that their transformer encoder-decoder structure leads to learning more object-specific information in its feature vectors and global scene context not being captured. As this global scene context is arguably important for VQA and other crossmodal tasks, we compensate for the loss in global context by *augmenting each output feature vector with context information from global average pooling*. We analyze different positions for extracting this contextual information and find a *substantial gain of 1.3% accuracy* for VQA<sub>v2</sub> from the global context. Full details are described in Sect. 4.4. While not completely closing the accuracy gap to Faster R-CNN features, we note that these are more high-dimensional and not amenable to end-to-end crossmodal training. Also note that adding the same contextual information to Faster R-CNN features, in contrast, does not aid VQA accuracy. Consequently, we conclude that *global contextual information about the scene is important for using transformer-based detectors for VQA*.

### 3.3 Scalable Deformable-DETR

Another significant impediment in the deployment of transformer-based architectures for multimodal tasks are the very significant training times of DETR [4]. This has been



**Fig. 1.** Number of extracted objects per image for COCO validation. Only predictions with a confidence exceeding a threshold are extracted, but a minimum of 10 and a maximum of 100 objects are selected per image. Thresholds are chosen such that the total number of extracted regions is the same for all methods. Deformable-DETR with focal loss produces a skewed distribution while using a cross-entropy loss resembles the detection distribution of Faster R-CNN and DETR.

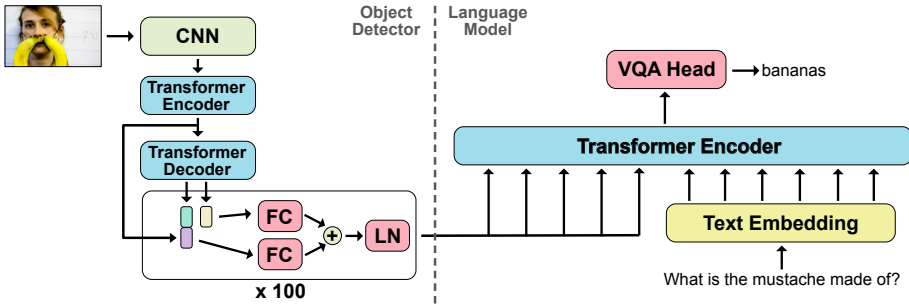
very recently addressed with Deformable-DETR [49], which is much faster to train, but is slower at test time, requiring more than double the computations compared to DETR. We address this through a *more scalable Deformable-DETR* method, which can still use the multi-scale information but is computationally more lightweight. Deformable-DETR extracts the feature maps of the last three ResNet blocks with strides of 8, 16, and 32. It also learns an additional convolution with stride 2 to produce a fourth feature map with stride 64. Those four feature maps are then fed to the multi-scale deformable attention layers of the encoder. In the deformable self-attention, each feature then attends to a fixed number of interpolated features of each of the feature maps, incorporating multi-scale information in the process.

Instead of performing a full self-attention, where query and keys are the same set, we use the feature maps of all scales as keys but propose to query only with the feature map of one chosen scale. This results in a *single feature map that still incorporates multi-scale information*, because the query features attend to the keys across all scales. All further attention layers then work on this single feature map, reducing the amount of computational operations needed. By choosing feature maps of different strides as the query, we can scale the computational complexity and accuracy corresponding to our task. See Appendix A in the supplemental material for detailed results.

For a good trade-off between computational cost and AP to be used for our cross-modal architecture, we query with feature maps of stride 16. In Table 1 we compare our model to Faster R-CNN with ResNet-50 and ResNet-101 backbones [12], DETR, DETR-DC5 (with dilated convolutions in the last ResNet block), and multi-scale Deformable-DETR. Compared to the standard DETR model, our proposed scalable Deformable-DETR maintains a comparable number of model parameters and reaches a *0.9% higher AP* with only 10% more computational expense while still being trained in only 50 instead of 500 epochs. It is also faster at test time than Faster R-CNN, while being more accurate when compared with the same backbone. Our approach thus provides a favorable trade-off as a detector basis.

### 3.4 Detection Losses for Crossmodal Tasks

Deformable-DETR not only aims to address the lacking training efficiency of DETR, but also its limitations for hard-to-detect objects, *e.g.* small objects or objects with rare



**Fig. 2.** Overview of our proposed TxT model: A transformer-based object detector [4, 49] produces feature representations and bounding box coordinates for its predictions (*left*). Following [5], each feature and position pair is combined by fully connected (FC) layers. The aggregated representations are passed through a layer norm (LN) and utilized as object embeddings; concatenated with the sequence of text embeddings, the multimodal instance is passed into the transformer-based language model (*right*). A task specific head (here for VQA) predicts the answer. The model is end-to-end trainable, including both the vision and language components.

classes. To that end, a focal loss [24] is employed. To assess if this impacts its use in downstream tasks, such as VQA, we analyze both the detection output as well as its use as pre-extracted feature for VQA. To compare the suitability of the features extracted by different object detectors, we only extract the features of objects with a confidence above a certain threshold. A minimum of 10 objects and a maximum of 100 objects per image are extracted following Anderson *et al.* [1]. The thresholds of DETR and Deformable-DETR are chosen such that all models produce the same total number of objects on the COCO validation set. When we plot the distribution of the number of objects per image across the validation dataset, shown in Fig. 1, we see that DETR, which is also trained with a cross-entropy loss, produces a distribution with a peak at a similar position as Faster R-CNN, although a little broader. Deformable-DETR trained with a focal loss to improve on difficult objects, on the other hand, has a distribution shifted to a lower number of regions and notably more spread out and skewed. Since small or rarely occurring objects (currently) do not play a significant role in VQA,<sup>3</sup> this property may not necessarily benefit the crossmodal task. Indeed, as shown in Table 2, the *focal loss degrades the VQAv2 performance*. Since we are less interested in pure detection accuracy, but rather in crossmodal performance, we therefore train our Deformable-DETR with a *cross-entropy loss for the class labels*. As can be seen in Fig. 1 (*right*), this makes the detection statistics more closely resemble those of Faster R-CNN and DETR, and leads to a *clear increase of 0.6% on VQAv2 test-dev* (*cf.* Table 2).

<sup>3</sup> Verified by measuring object sizes via gradient backtracing in UNITER [5] using the Grad-CAM [39] approach and selecting image regions with normalized activations greater than 0.9.



### 3.5 The TxT Model

With the proposed *TxT model*, we now combine transformer-based object detection (T) and transformer-based language models (T) in a crossmodal approach (x). It enables us to bridge the information gap between language and vision models and perform cross-modal end-to-end training while avoiding computationally heavy dense features [16]. We base the language model on BERT-base, pre-trained with MLM on text corpora only [6]. For the object detector, we employ either DETR or Deformable-DETR models as described in Sects. 3.1 to 3.4. The structure of TxT is shown in Fig. 2. In the object detector a CNN backbone (ResNet-50 [12]) computes feature representations from the input image. They are passed to the transformer encoder, where they interact in its multi-head attention layers. In the decoder, the encoded feature map gets queried for objects and for each object the feature representation and the bounding box coordinates are extracted [4] (see Sect. 3.3 for details). The global average-pooled encoder output is concatenated to the feature representation of each object to provide global context (*cf.* Sect. 3.2). We use the complete set of predictions generated by the object detector for TxT. Following [5], features and bounding box coordinates are projected by fully connected layers to the dimension of the BERT model and combined. After passing a normalization layer, they are used as object embeddings in the language model.

The text input is run through a tokenizer and both embeddings, visual and textual, are concatenated to a sequence so that the self-attention of the transformer encoder is able to reason over both modalities. The encoder output is passed through a task-specific head, in our case a VQA head, consisting of a small multi-layer perceptron (MLP) to produce the answer predictions, again following [5]. The TxT model is trained fully end-to-end with a binary cross-entropy loss across the border between the vision and the language part. Thus, we are able to fine-tune the object detector specifically for the crossmodal task.

## 4 Experiments

To evaluate our TxT model, we perform experiments on a visual questioning answering task. Specifically, we use the VQAv2 training set [11] and add additional question and answer pairs from Visual Genome [20] for augmentation, following [1, 5], but omit using the validation split for training. We follow standard practice and only classify for the 3129 most frequent answers [1, 5, 47]. All results are evaluated with the standard VQA accuracy metrics [2] on the test-dev set from the 2021 challenge for VQAv2 [11].

### 4.1 Technical Details

*Object Detector.* The transformer-based object detectors in our TxT model employ a ResNet-50 [12] backbone. The extracted features are projected to a hidden dimension of 256 of the transformer layers and then passed to an encoder consisting of 6 transformer layers. The transformer layers consist of multi-head attention layers with 8 heads and feed-forward networks with a dimension of 2048 for DETR and 1024 for Deformable-DETR, as described in [45]. Learned object queries generate 100 object features with

an alternation of self-attention and cross-attention in the 6 transformer layers of the decoder. A linear projection is used to classify the object and a 3-layer perceptron is used to generate bounding box coordinates [4]. Deformable-DETR uses 300 queries in its default configuration. For end-to-end learning in TxT, we use a variant with 100 object queries. Deformable-DETR uses multi-scale deformable attention in its transformer layers, which employs 4 keys interpolated around each reference point of the query. We use Deformable-DETR with iterative bounding box refinement [49]. Both DETR and Deformable-DETR use global context as discussed in Sect. 3.2.

*Pre-training.* Following Anderson *et al.* [1], we pre-train all object detectors on the Visual Genome dataset [20] with a maximum input image size of  $600 \times 1000$  and add an additional head for the prediction of object attributes. We train DETR for 300 epochs on 8 NVIDIA V100 GPUs with 4 images per GPU. We use a learning rate of  $1e-4$  for DETR and a learning rate of  $1e-5$  for the ResNet backbone. After 200 epochs, we drop the learning rate by a factor of 10. For all other settings, we follow [4]. The training protocol for Deformable-DETR follows [49].

*Language Model.* The language model of TxT is based on BERT-base [6] with 12 transformer layers and 768 hidden dimensions. BERT is pre-trained on the combination of the BooksCorpus [50] and the entire English Wikipedia corpus, consisting of 3.3 billion words in total. A WordPiece tokenizer [46] is trained on the entire corpus, amounting to a vocabulary size of 30k, with each token corresponding to a position in the designated word-embedding matrix. *Positional* embeddings are trained to preserve the sentence order, and *segment* embeddings are trained to indicate if a token belongs to sentence A or B. The embedding representation, which is passed into the transformer, thus amounts to the sum over *word*-, *positional*-, and *segment*-embedding of each respective token. BERT is pre-trained with an MLM,<sup>4</sup> as well as a next-sentence prediction objective.<sup>5</sup>

*VQA Training.* The TxT model is trained for 10k iterations on 4 GPUs and a learning rate of  $8e-5$ . We employ a binary cross-entropy loss and the AdamW [28] optimizer. We use a linear warmup of 600 iterations and a linear decay of the learning rate afterwards. In case of pre-extracted, fixed object features, we train the TxT model similar to [5] with 5120 input tokens and for each iteration we accumulate the gradients over 5 steps. For end-to-end training, we pre-train the TxT model with fixed features for 4k iterations for DETR and 6k iterations for Deformable-DETR with the above settings. Then, we train the model fully end-to-end for 10k iterations with 8 images and questions per batch and accumulate the gradients for 32 steps for each iteration to maintain the effective batch size of the fixed-feature setting. When we employ TxT with DETR as the object detector, we initialize the learning rate for the DETR and CNN backbone components with  $3e-4$  and  $3e-5$ , respectively. Deformable-DETR is trained with a learning

<sup>4</sup> Input tokens are randomly masked, with the objective of the model being to predict the missing token given the surrounding context.

<sup>5</sup> Two sentences A and B are passed into the model, where 50% of the time B follows A in the corpus, and 50% of the time B is randomly sampled. The objective is to predict whether or not B is a negative sample.

**Table 2.** Comparison of VQA accuracy on pre-extracted features from different object detectors evaluated on VQAv2 test-dev [11] (higher is better). Deformable-DETR refers to our variant, where we query the attention module with a feature map of stride 16 (see Sect. 3.3 for details). The best results in each column are bold, the 2<sup>nd</sup> best underlined.

Method	number	yes/no	other	overall
Faster R-CNN	<b>51.18</b>	<b>85.04</b>	<b>59.00</b>	<b>68.84</b>
DETR	49.49	83.88	<u>57.81</u>	67.60
Deformable-DETR (focal loss)	47.59	83.87	57.18	67.09
Deformable-DETR (cross-entropy loss)	48.66	<u>84.46</u>	57.62	<u>67.66</u>

rate of  $6e-4$  and  $6e-5$  for the backbone. When TxT is used with DETR and thresholding the predictions, we apply a factor of 10 to the learning rate of the class-prediction layer of DETR. For end-to-end training of TxT, the learning rates and training schedule were determined with a coarse grid search over the hyperparameters.

## 4.2 Representational Power of Visual Features

In a first step, we investigate if DETR and Deformable-DETR are able to produce object feature descriptors whose representational power for VQA tasks is comparable to the widely used Faster R-CNN; we assess this through the VQA accuracy. We pre-extract object features and bounding box coordinates with all object detectors. Following standard practice (e.g. [1, 5, 42]), all objects with a class confidence above a certain threshold are extracted, but a minimum of 10 and a maximum of 100 objects are used per image. We calibrate the confidence threshold of DETR and Deformable-DETR on COCO validation so that the total number of extracted object features is 1.3 million, comparable to Faster R-CNN. The TxT model is then trained with the pre-extracted features. When used with Faster R-CNN features, this is equivalent to the UNITER-base model [5] with only MLM pre-training, i.e. initialized with BERT weights. We note that the numbers differ slightly from those in [5], because we omit the validation set when training for test-dev submission and use the most recent VQA challenge 2021.

The results in Table 2 show that DETR and Deformable-DETR are able to produce object features with a representational power that *competitively supports VQA tasks*, especially considering their low dimensionality of 256 (compared to 2048 dimensions of Faster R-CNN features). Also, these results are achieved with only a ResNet-50 backbone (compared to ResNet-101 for Faster R-CNN). While DETR leads to a  $\sim 1.2\%$  loss in VQA accuracy compared to Faster R-CNN, it is more efficient at test time (see Table 1) and, more importantly, allows for end-to-end training (cf. Sect. 4.3).

Moreover, we find that Deformable-DETR (with query stride 16) as the object detector for VQA performs less well than DETR, about 0.5% worse in VQA accuracy. As analyzed in Sect. 3.4, this is traceable to the focal loss. We find that a cross-entropy loss is a more suitable pre-training objective for Deformable-DETR as object detector for the VQA task, leading to  $\sim 0.6\%$  higher accuracy on VQA while maintaining the same total number of predictions. With these modifications, *Deformable-DETR performs comparatively with DETR, yet is still much faster to train.*

**Table 3.** Results of the multimodal end-to-end trainable TxT model on VQAv2 test-dev [11] (higher is better). *e2e* denotes end-to-end training. In the upper part of the table, we show the results of counterparts with pre-extracted features and Faster R-CNN features. The best results in each column are bold, the 2<sup>nd</sup> best underlined.

Method	e2e	number	yes/no	other	overall
Faster R-CNN		51.18	85.04	59.00	68.84
DETR		49.49	83.88	57.81	67.60
DETR (all predictions)		<b>52.33</b>	85.57	<u>59.84</u>	<u>69.59</u>
Deformable-DETR (100 queries)		47.19	84.05	56.92	66.99
Deformable-DETR (100 q., all predictions)		48.54	83.79	57.82	67.47
TxT-DETR (thresholded)	✓	50.59	85.66	59.23	69.14
TxT-DETR	✓	<u>51.73</u>	<b>86.33</b>	<b>60.04</b>	<b>69.93</b>
TxT-Deformable-DETR	✓	49.70	<u>85.76</u>	59.19	69.06

### 4.3 Multimodal End-to-End Training

As a preliminary experiment, referred to as TxT-DETR (thresholded), we only pass objects to the multimodal transformer layers where the class confidence exceeds a threshold of 0.5. To accomplish this, we generate a mask from thresholding the class confidence. We then multiply the object features with the mask, thus allowing the gradient to backpropagate to the class prediction layer of DETR. Only features and related bounding box coordinates according to the mask are then passed from the DETR part of TxT to its language model. At the beginning of training, DETR selects around 30 objects per image, equivalent to the setting of pre-extracted features. During training, the TxT model learns to select more objects, saturating at around 80 per image. This suggests that *more objects are beneficial for solving the VQA task*, which confirms an empirical observation of Jiang *et al.* [17], who showed that the accuracy starts to saturate between 100 and 200 regions. As Table 3 shows, including *all* predicted objects in the multimodal reasoning (DETR *all predictions*) without end-to-end training indeed leads to a  $\sim 0.5\%$  higher accuracy than TxT-DETR (thresholded), which leverages the end-to-end training.

For multimodal end-to-end training of our TxT model, we therefore employ all 100 object predictions, eliminating the need to threshold and enabling a gradient for object predictions that would be discarded otherwise. In order to alleviate the computational and memory cost of multimodal self-attention, we pre-train Deformable-DETR with only 100 object queries (instead of the standard 300), which reduces the VQA accuracy by  $\sim 0.7\%$  when evaluating in the pre-extracted features setup. To make the effect of the higher object number distinguishable from the end-to-end benefit, we also show results for pre-extracted features training with all object predictions in Table 3. With DETR as object detector, the gain from using all predictions is  $\sim 2\%$  and for Deformable-DETR it is less pronounced with a benefit of  $\sim 0.5\%$ .

Finally, we assess our full end-to-end trainable model. We find that *multimodal end-to-end training substantially improves the accuracy of TxT*: The TxT model with

**Table 4.** *Global context features* for DETR [4] and Faster R-CNN [37] from different locations in the network. The results on a VQA task are evaluated on VQAv2 [11] test-dev (in %, higher is better). We also report the total dimensionality  $d$  of the extracted features.

Model	$d$	number	yes/no	other	overall
DETR (no context)	256	47.65	82.67	56.61	66.33
DETR (backbone context)	2304	48.80	83.68	57.38	67.24
DETR (projected backbone context)	512	48.69	83.43	57.36	67.12
DETR (encoder context)	512	49.49	83.88	57.81	67.60
Faster R-CNN (no context)	2048	51.18	85.04	59.00	68.84
Faster R-CNN (backbone context)	3072	49.86	84.29	58.37	68.08

Deformable-DETR improves by  $\sim 2.1\%$  from 66.99% for pre-extracted Deformable-DETR features to 69.06% accuracy on VQAv2 test-dev. The TxT model in combination with DETR achieves an overall gain in accuracy of  $\sim 2.3\%$  from 67.60% accuracy for pre-extracted DETR features to 69.93% accuracy for multimodal end-to-end training, thus improving by  $\sim 1.1\%$  over the Faster R-CNN features. While our results are competitive with current models on an equal footing, they could be improved further with pre-training tasks like image-text matching and word-region alignment [5] at the expense of significant computational overhead.

#### 4.4 Ablation Study

As discussed in Sect. 3.2, adding global context is helping DETR [4] to produce visual features that are better suited for VQA. We now investigate the suitable position from which to source this global information. We consider three positions at which to apply a global average pooling of the feature map in the DETR network: (i) pooling the feature map produced by the CNN backbone, (ii) after the backbone feature map is projected to the lower dimensions of the transformer layers, and (iii) pooling the encoder output. As we can see in Table 4, *all variants with added global context lead to a gain in accuracy*. We obtain the best results when using the *encoder output*, leading to a gain of  $\sim 1.3\%$  in terms of VQAv2 [11] accuracy over using no global context. Pooling the feature map produced by the CNN backbone gives a slightly lower gain of  $\sim 0.9\%$  and requires 2304-dimensional features compared to only 512 dimensions for the encoder output. Projecting the backbone feature map first to the transformer dimension also gives 512-dimensional features, but yields only a  $\sim 0.8\%$  gain.

To ensure a fair comparison, we also verify if Faster R-CNN [37] features similarly benefit from global context. To that end, we concatenate the Faster R-CNN features with global average-pooled features of its shared backbone. In contrast to DETR, this does not lead to better VQA results, rather to a slight degradation in accuracy of  $\sim 0.7\%$ . We conclude that global context is important for VQA, yet is sufficiently represented in the common Faster R-CNN features. Transformer-based detectors, in contrast, benefit significantly from the added context.

## 5 Conclusion

In this paper we proposed TxT, an end-to-end trainable transformer-based architecture for crossmodal reasoning. Starting from the observation that transformer-based architectures have not only been successfully used for language modeling, but more recently also for object detection, we investigated the use of Detection Transformers and variants as source of visual features for visual question answering. We found that global context information as well as an adaptation of the loss function are needed to yield rich visual evidence for crossmodal tasks. We also proposed a speed-up mechanism for multi-scale attention. Our final end-to-end trainable architecture yields a clear improvement in terms of VQA accuracy over standard pre-trained visual features and paves the way for a tighter integration of visual and textual reasoning with a unified network architecture.

**Acknowledgement.** This work has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center.

## References

1. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR, pp. 6077–6086 (2018)
2. Antol, S., et al.: VQA: visual question answering. In: ICCV, pp. 2425–2433 (2015)
3. Brown, T.B., et al.: Language models are few-shot learners. [arXiv:2005.14165](https://arxiv.org/abs/2005.14165) [cs.CL] (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13)
5. Chen, Y.-C., et al.: UNITER: UNiversal image-TExt representation learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 104–120. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58577-8\\_7](https://doi.org/10.1007/978-3-030-58577-8_7)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
7. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. In: ICLR (2021)
8. Gan, Z., Chen, Y., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. In: NeurIPS, pp. 6616–6628 (2020)
9. Girshick, R.B.: Fast R-CNN. In: ICCV, pp. 1440–1448 (2015)
10. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 142–158 (2016)
11. Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.* **127**(4), 398–414 (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)

14. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: ACL, pp. 328–339 (2018)
15. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR, pp. 3296–3297 (2017)
16. Huang, Z., Zeng, Z., Liu, B., Fu, D., Fu, J.: Pixel-BERT: aligning image pixels with text by deep multi-modal transformers. [arXiv:2004.00849](https://arxiv.org/abs/2004.00849) [cv.CV] (2020)
17. Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E.G., Chen, X.: In defense of grid features for visual question answering. In: CVPR, pp. 10264–10273 (2020)
18. Kamath, A., Singh, M., LeCun, Y., Misra, I., Synnaeve, G., Carion, N.: MDETR - modulated detection for end-to-end multi-modal understanding. [arXiv:2104.12763](https://arxiv.org/abs/2104.12763) [cs.CV] (2021)
19. Kiela, D., Bhooshan, S., Firooz, H., Testuggine, D.: Supervised multimodal bitransformers for classifying images and text. In: Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop (2019)
20. Krishna, R., et al.: Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **123**(1), 32–73 (2017)
21. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL, pp. 7871–7880 (2020)
22. Li, X., et al.: Oscar: object-semantics aligned pre-training for vision-language tasks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 121–137. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58577-8\\_8](https://doi.org/10.1007/978-3-030-58577-8_8)
23. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, pp. 936–944 (2017)
24. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(2), 318–327 (2020)
25. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
26. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
27. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) [cs.CL] (2019)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
29. Lu, J., Batra, D., Parikh, D., Lee, S.: ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: NeurIPS, pp. 13–23 (2019)
30. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2Text: describing images using 1 million captioned photographs. In: NIPS, pp. 1143–1151 (2011)
31. Peters, M., Ammar, W., Bhagavatula, C., Power, R.: Semi-supervised sequence tagging with bidirectional language models. In: ACL, pp. 1756–1765, July 2017
32. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV, pp. 2641–2649 (2015)
33. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Technical report, OpenAI (2018)
34. Radford, A., Wu, J., R., C., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. Technical report, OpenAI (2019)
35. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
36. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)

37. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
38. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
39. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **128**(2), 336–359 (2020)
40. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: a cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: *ACL*, pp. 2556–2565, July 2018
41. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017)
42. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: VL-BERT: pre-training of generic visual-linguistic representations. In: *ICLR* (2020)
43. Tan, H., Bansal, M.: LXMERT: learning cross-modality encoder representations from transformers. In: *EMNLP-IJCNLP*, pp. 5099–5110 (2019)
44. Teney, D., Anderson, P., He, X., van den Hengel, A.: Tips and tricks for visual question answering: learnings from the 2017 challenge. In: *CVPR*, pp. 4223–4232 (2018)
45. Vaswani, A., et al.: Attention is all you need. In: *NIPS*, pp. 5998–6008 (2017)
46. Wu, Y., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) cs.[CL] (2016)
47. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: *CVPR*, pp. 6281–6290 (2019)
48. Yu, Z., Yu, J., Xiang, C., Fan, J., Tao, D.: Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(12), 5947–5959 (2018)
49. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: *ICLR* (2021)
50. Zhu, Y., et al.: Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: *ICCV*, pp. 19–27 (2015)