Recurrent Glimpse-based Decoder for Detection with Transformer

Zhe Chen[1] Jing Zhang[1] Dacheng Tao[2,1]
[1] The University of Sydney, Australia [2] JD Explore Academy, China
{zhe.chen1,jing.zhang1}@sydney.edu.au; dacheng.tao@gmail.com

Abstract

Although detection with Transformer (DETR) is increasingly popular, its global attention modeling requires an extremely long training period to optimize and achieve promising detection performance. Alternative to existing studies that mainly develop advanced feature or embedding designs to tackle the training issue, we point out that the Region-of-Interest (RoI) based detection refinement can easily help mitigate the difficulty of training for DETR methods. Based on this, we introduce a novel REcurrent Glimpse-based decOder (REGO) in this paper. In particular, the REGO employs a multi-stage recurrent processing structure to help the attention of DETR gradually focus on foreground objects more accurately. In each processing stage, visual features are extracted as glimpse features from RoIs with enlarged bounding box areas of detection results from the previous stage. Then, a glimpse-based decoder is introduced to provide refined detection results based on both the glimpse features and the attention modeling outputs of the previous stage. In practice, REGO can be easily embedded in representative DETR variants while maintaining their fully end-to-end training and inference pipelines. In particular, REGO helps Deformable DETR achieve 44.8 AP on the MSCOCO dataset with only 36 training epochs, compared with the first DETR and the Deformable DETR that require 500 and 50 epochs to achieve comparable performance, respectively. Experiments also show that REGO consistently boosts the performance of different DETR detectors by up to 7% relative gain at the same setting of 50 training epochs. Code is available via https://github.com/zhechen/Deformable-DETR-REGO.

1. Introduction

Object detection aims to locate and recognize foreground objects from images. In recent years, deep learning has made rapid development in object detection. With deep convolutional neural networks [18, 19, 27, 33, 46], various powerful detectors have been developed [4, 22, 32, 37].

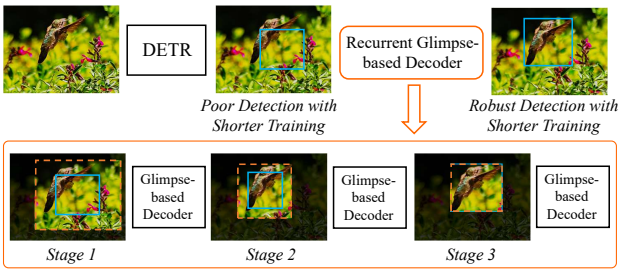

Figure 1. Concept of the proposed recurrent glimpse-based decoder (REGO) for augmenting the training of attention modeling in Detection with Transformer (DETR). Using original DETR results, the REGO performs a multi-stage Region-of-Interest (RoI) based attention modeling refinement procedure by gradually focusing on more accurate areas. In each stage, glimpse features are extracted and a glimpse-based decoder is employed to provide refined detection outputs based on both the glimpse features and the attention modeling output of the previous stage. The REGO maintains the fully end-to-end pipeline of different DETR methods and can improve their training performance promisingly.

In general, modern detectors produce redundant results and require Non-Maximum Suppression (NMS) to reduce the redundancy in detection. Different from this popular paradigm, Detection with Transformer (DETR) [3] applied Transformer [39] for detection and is the first fully end-to-end detector that avoids the need for NMS. In particular, a Transformer is a powerful attention-based encoder-decoder pipeline for translating an input sequence to the target sequence. By formulating the detection task as a direct set prediction problem, the authors of DETR managed to translate visual features into a set of detection results based on the global attention modeling of a Transformer. Despite benefits, the DETR suffers from a difficult training problem. Using MS COCO dataset [24], the original DETR requires 500 training epochs to obtain promising performance, while the other popular detectors like FPN [22] only require less than 36 epochs to get similar results. Even using a machine with 8 powerful V100 GPUs, a DETR detector costs more than 10 days to finish the training [3].

By addressing the training problem, researchers found

that the lack of effective locality modeling could affect the training of attention modeling in DETR methods. For example, Zhu *et al.* [48] analyzed that the Transformer would distribute almost uniform attentional weights to all features initially. It is then necessary to apply long training epochs to make the Transformer learn to focus on sparse and meaningful local areas. To tackle this issue, researchers developed advanced multi-scale feature encoding [9, 12] and object embedding designs [28, 41] to improve the locality modeling in Transformer before final detection, so that the attention of Transformer can be trained more efficiently and the detection results can be improved properly.

Different from existing methods, we propose that the training of the attention modeling in DETR can be easily improved based on Region-of-Interest (RoI). More specifically, considering local areas around bounding boxes detected by DETR as RoIs that may contain objects, we can directly restrict the attention of DETR by only focusing on these RoIs. Therefore, modeling the features within RoIs can help introduce more locality inductive biases in DETR and thus improve its training efficiency effectively. In fact, researchers have demonstrated that gradual refinements according to RoIs can boost training and detection performance for two-stage [15, 32] and multi-stage detectors [2, 32]. Nevertheless, these multi-stage detection methods mainly follow RCNN detection methodology [16] for training and inference which still requires NMS. To our best knowledge, the RoI-based refinement for attention modeling in DETR has rarely been studied.

To develop a proper RoI-based DETR refinement method, we take inspiration from the glimpse mechanism as studied in the work [29] which extracted features from a few selected local areas of different scales as glimpses and applied a recurrent network to encode the glimpse information. Similar to DETR, this glimpse method also formulates the visual understanding as a sequence translation task and has proven to be effective for image recognition. We follow this mechanism and propose a novel recurrent glimpse-based decoder (REGO) module to help existing DETR methods relieve training difficulty and improve detection performance.

The proposed REGO module refines DETR with multi-stage processing. Taking detection and attention modeling outputs of the original DETR as initial states, each stage of REGO first extracts glimpse features from local areas surrounding the detected bounding boxes. Then, a Transformer decoder is employed to translate the glimpse features based on previous attention modeling outputs into augmented attention modeling outputs and refined detection results. For early stages, we extract glimpse features from the local areas at larger scales *w.r.t.* the detected bounding box areas, enabling the incorporation of rich contexts to boost the detection that can possibly be unreliable in early stages. After multiple stages of processing, the REGO performs a coarse-to-fine RoI-based refinement which is shown to be effective for improving the training of different DETR methods.

To sum up, the contributions of this paper are three-fold:

- We proposed a novel RoI-based refinement module that can effectively tackle the difficult training problem for the attention modeling in DETR and improve detection performance.
- The REGO is easy-to-implement and is a complementary module that can be embedded in different DETR variants. It keeps the fully end-to-end detection pipeline of DETR while accelerating convergence and improving detection performance for different DETR methods effectively.
- Extensive experiments show that the REGO helps deliver promising performance using only 36 training epochs with a DETR pipeline, which is 13× shorter than the first DETR method. Moreover, REGO also consistently boosts the performance of different DETR methods by up to 7% relative gain using the same 50 training epochs.

2. Related Work

Object Detection Modern detectors [7, 8, 17, 22, 32] generally make dense detection of objects appeared on the images. For example, the widely used regional proposal network (RPN) [32] scans every location on the feature map of the backbone like ResNet [18] and generates proposal windows that may cover foreground objects. This produces plenty of redundant proposals, *e.g.*, an object can be covered by different but highly overlapped proposals, which is disadvantageous to make sparse predictions. To alleviate this problem, one-stage methods [21, 23, 26, 31] develop augmented networks to ensure that they can directly provide compact detection results. Two-stage methods [17, 32] attempt to refine the proposal bounding boxes based on the features extracted with RoIPooling [15] or RoIAlign [17]. Nevertheless, both one-stage and two-stage detectors rely on the hand-designed NMS procedure to remove redundancy, which is heuristic and separated from the end-to-end learning pipeline, leading to many inaccurate predictions remained after NMS. Alternatively, recently introduced detection with Transformer [3] can provide a set of object detection results without requiring NMS. However, DETR suffers from frustratingly difficult training problem.

Improvement of Transformer in Computer Vision The training difficulty of DETR is a common issue in Transformer-based computer vision methods [11, 38, 40]. By addressing the training problem, many researchers found that locality modeling is important for improving the

training of attention in DETR [36, 43, 47]. Some methods [30] developed advanced local window-based method for improving efficiency. In object detection, researchers mainly develop advanced feature encoding and embedding designs to help tackle this problem. The Deformable DETR [48] applied deformable operations [9] to better focus on a few local areas at different scales in the Transformer. The method SMCA [12] introduced multi-scale co-attention to improve DETR with refined local representations. In addition, other studies like Conditional DETR [28] and Anchor DETR [41] tend to improve the spatial embedding in Transformer to help accelerate training. These two methods enhance the locality modeling of Transformer by making attention focus on potentially valuable areas on the image learned with positional embeddings. Unlike these methods that require careful designs, we argue that the RoIs which naturally correspond to local areas can also improve the training of attention modeling in DETR. A more related method is the iterative refinement used in Deformable DETR [48]. We note that this method does not use RoIs and it mainly improves performance by re-using all the regression outputs of DETR. Our RoI method can improve attention modeling and is orthogonal to this method. Experiments show that the cooperation of this method and our REGO achieves state-of-the-art performance.

RoI-based Improvement for Object Detection Researchers have proved that the detection results can be progressively improved by refining classification and localization *w.r.t.* RoIs [2, 6, 14, 25, 44]. For example, MR CNN [13] introduced an iterative procedure to alternate between scoring and bounding box refinement based on RoIs. The CascadeRCNN [2] repeated the RoI-based detection head of the Faster RCNN [32] several times for refinement. Despite effectiveness, such type of RoI-based refinement methodology can not be directly applied to the fully end-to-end pipeline of DETR because they rely on different optimization goals and still require NMS. More recently, some methods, like Efficient DETR [45], TSP-RCNN [35], and SparseRCNN [34], also uses RoIs to achieve improved performance with a Transformer and can also avoid the NMS. However, we argue that these methods are still based on the typical two-stage detection pipeline like Faster RCNN [32] and they only apply Transformer mainly to approximate NMS. These methods do not directly tackle the difficult training problem for attention modeling in DETR.

In summary, exploring end-to-end RoI-based refinement for improving the training of attention modeling in DETR remains a missing part in literature.

3. Preliminary

Here, we briefly review the DETR. More details can be found at [3, 39].

Multi-head Attention Multi-head attention deals with *query*, *key*, and *value* inputs. It correlates query and key and then aggregates values according to the correlation results. Following [39], the multi-head attention splits features into different 'heads' and performs self-attention or cross-attention in each head. The features of different heads will be concatenated together and fed into a linear projection to obtain the final output. Formally, to help describe a multi-head attention, we suppose that $X_q \in \mathbb{R}^{L_q \times C}$ is a query tensor where L_q refers to its sequence length and C is its feature dimension. We follow the formulations of DETR [3] and unify the key and value into the same tensor: $X_{kv} \in \mathbb{R}^{L_{kv} \times C}$ which is the the key-value sequence of length L_{kv}. The multi-head attention, abbreviated as "$\mathcal{A}$", can be formulated as:

$$\mathcal{A}(X_q, X_{kv}) = W_{\mathcal{A}}\big[\mathcal{A}(X_q^1, X_{kv}^1), \dots, \mathcal{A}(X_q^M, X_{kv}^M)\big], \tag{1}$$

where $W_{\mathcal{A}} \in \mathbb{R}^{C \times C}$ is a trainable linear projection matrix, M is the number of heads, and $[\dots]$ refers to concatenation operation. The $X_q^i \in \mathbb{R}^{L_q \times C'}$ and $X_{kv}^i \in \mathbb{R}^{L_{kv} \times C'}$ are query and key-value tensors of the i-th head ($i = 1, \dots M$), respectively, where $C' = \frac{C}{M}$. In each head, the following operation is performed:

$$\mathcal{A}(X_q^i, X_{kv}^i) = \mathcal{A}_{qkv}^i X_{kv}^i, \tag{2}$$

where $\mathcal{A}_{qkv}^i$ represents the attentional weights: $\mathcal{A}_{qkv}^i = Softmax(\frac{X_q^i (X_{kv}^i)^T}{\sqrt{C'}})$.

DETR Pipeline The DETR applies an encoder-decoder pipeline to translate the input features into a set of detection results. During training, Hungarian matching [20] is performed to assign the detection results with the most matched ground-truths. The encoder-decoder consists of a visual feature encoding phase and a detection result decoding phase. The feature encoding investigates the relations between visual features from different locations. It applies several encoding layers to augment the encoded representation. We suppose the backbone network extracts features into: $X \in \mathbb{R}^{H \times W \times C}$ where H, W represents the height and width, respectively, and C is the feature dimension. In each encoding layer, a multi-head *self-attention* module is employed, that is, query, key, and value tensors are the same: $X_q = X_{kv}$. The input feature X also integrates a positional embedding to encode position information. Suppose the output of the encoding phase is $H_{enc} \in \mathbb{R}^{HW \times C}$. Then, detection decoding phase perform detection based on H_{enc}. It begins from object query embeddings $E_{box} \in \mathbb{R}^{N_d \times C}$ and applies *cross-attention* as described in Eq. (1) with H_{enc} for making predictions. The N_d here represents the number of predicted objects. Suppose the decoded feature is H_{dec}, then $H_{dec} = \mathcal{A}(E_{box}, H_{enc})$. With the H_{dec}, the decoding phase performs classification and bounding box coordinate regression, obtaining $O_{cls} \in \mathbb{R}^{N_d \times N_c}$ and $O_{box} \in \mathbb{R}^{N_d \times 4}$,

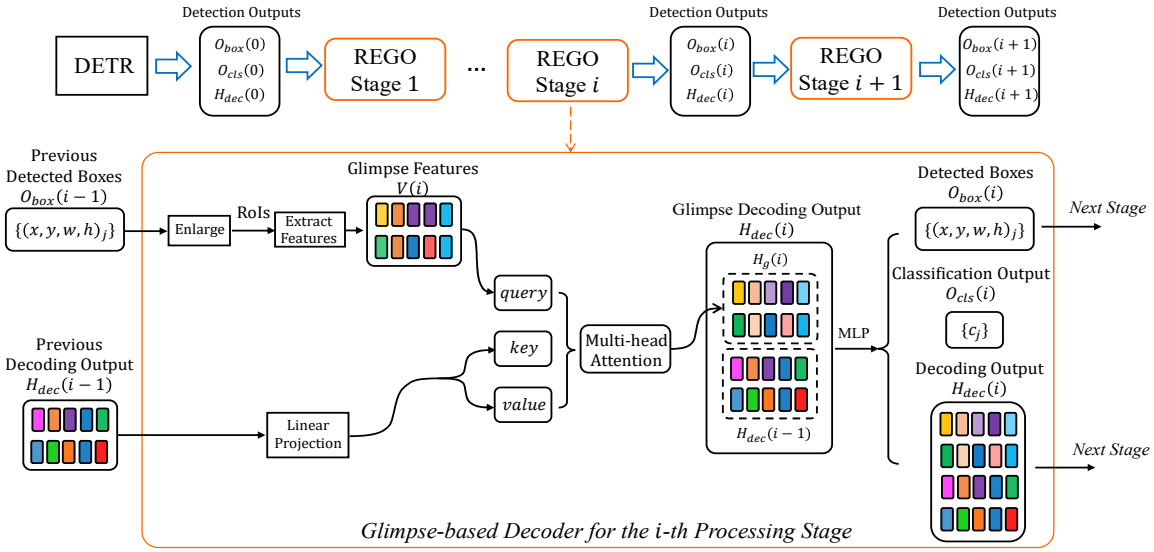

Figure 2. The overview of the REGO (top row) and the detailed structure of the i-th processing stage in REGO.

respectively, where N_c represents the number of object categories. To this end, we have:

$$\begin{cases} O_{cls} = F_{cls}(H_{dec}) \\ O_{box} = F_{box}(H_{dec}) \end{cases}, \tag{3}$$

where F_{cls} and F_{box} are functions that map the decoded feature H_{dec} into the desired outputs respectively. The two functions are implemented based on linear projection and multi-layer perception, respectively.

4. Recurrent Glimpse-based Decoder

Different from existing methods, we propose a recurrent glimpse-based decoders (REGO) to perform RoI-based detection refinement method for improving attention modeling in DETR. The REGO consists of two major components. The first one is a multi-stage recurrent processing structure that progressively augments attention modeling outputs and improves the detection of DETR, and the second one is the glimpse-based decoder that is used in each stage to explicitly perform the refinement. Figure 2 shows the detailed pipeline.

4.1. Multi-stage Recurrent Processing

Built upon detection results and attention decoding outputs from original DETR, we propose a recurrent processing pipeline to help the DETR gradually attend to more meaningful areas to avoid long training periods for optimizing the attention of DETR. In general, the proposed recurrent processing structure is a multi-stage pipeline. In each stage, previously detected bounding boxes are used to obtain RoIs for extracting glimpse features. Then, glimpse features are translated according to previous attention decoding outputs into refined attention decoding outputs for describing detected objects. The refined attention decoding outputs can provide improved detection results. Thus, for the i-th processing stage, we propose to detect objects according to:

$$\begin{cases} O_{cls}(i) = F_{cls}(H_{dec}(i)) \\ O_{box}(i) = F_{box}(H_{dec}(i)) + O_{box}(i-1) \end{cases}, \tag{4}$$

where $O_{cls/box}(i)$ represent the classification and bounding box regression outputs of the i-th recurrent processing stage, respectively, and $H_{dec}(i)$ represents the refined attention of this stage after decoding. Then, to obtain a proper representation of $H_{dec}(i)$, we use the following formulation:

$$H_{dec}(i) = [H_g(i), H_{dec}(i-1)], \tag{5}$$

where $H_g(i)$ is the translated glimpse features according to $H_{dec}(i-1)$, and $[\ldots]$ refers to concatenation operation. Reusing $H_{dec}(i-1)$ in Eq. (5) not only improves the attention of previous stages, but also help maintain consistency in the produced detection results across different stages, which could help reduce the variations in the Hungarian matching loss in later stages The study [35] has proven that reducing the randomness of the matching loss is beneficial for accelerating convergence. The calculation of translated glimpse features $H_g(i)$ will be discussed with more details in the next section. For the first stage where $i = 0$, we use the outputs of original DETR, as described in Eq. (3), to represent $O_{cls}(0)$, $O_{box}(0)$, and $H_{dec}(0)$.

4.2. Glimpse-based Decoder

During the i-th processing stage, the glimpse-based decoder collects visual features from areas around the detected bounding boxes $O_{box}(i-1)$ from the previous stage. It then

performs cross-attention to model the relations between the collected features and previous attention outputs and compute translated glimpse features $H_g(i)$ of current stage.

In particular, we denote the extracted visual features as $V(i)$ for the i-th stage, terming as the glimpse feature. Then, we translate it according to the previous attention outputs into a refined attention modeling outputs for detection. Multi-head cross-attention is applied to fulfill the translation, *i.e.*,

$$H_g(i) = \mathcal{A}(V(i), H_{dec}(i-1)). \tag{6}$$

Note that we use the attention outputs from the last layer of the decoder in original DETR to define $H_{dec}(0)$. It is also worth mentioning that either the $V(i)$ or the $H_{dec}(i-1)$ can be used as the query in $\mathcal{A}$. Both settings can correlate glimpse features with previous attention outputs properly and can all improve the training of DETR. We simply found that above formulation achieves 0.5 point higher in AP on COCO dataset [24].

To extract the glimpse features $V(i)$, we perform the following operation based on $O_{box}(i-1)$:

$$V(i) = f_{ext}\Big(X, R\big(O_{box}(i-1), \alpha(i)\big)\Big), \tag{7}$$

where the function f_{ext} represents the feature extraction operation, R represents the RoI computation, and $\alpha(i)$ a scalar factor. In particular, the function R computes RoIs by enlarging the areas of bounding boxes detected by $O_{box}(i-1)$ with a factor of α. Then, we use the RoIAlign [17] technique to implement f_{ext}. The symbol X here represents the features obtained with the backbone network.

Since the original detection results could be unreliable at first, we tend to extract glimpse features from a larger area around each detection result for refinement in early stages, so that contexts can be incorporated and target objects can be properly captured within the glimpse areas. In later processing stages, we gradually narrow the area for extracting glimpse features to achieve more precise detection with more local details. In other words, the $\alpha(i)$ in REGO starts from a large number and then decreases its value for later stages of the REGO. The detailed setting of $\alpha(i)$ can be found in the following section.

4.3. Implementation Details

The proposed REGO is a plug-and-play module for different DETR methods. It only has two major hyperparameters, *i.e.* the number of recurrent stages and the enlarging ratios α of each stage. To reduce the manual tuning efforts, we unify the two hyperparameters into a single one. More specifically, we constrain that the last recurrent stage has an enlarging ratio equals to 1. Then, when we add a new recurrent stage before the last stage, we increase the enlarging ratio by 1 for the added stage. In other words, if we use 3 recurrent stage, then $\alpha(3), \alpha(2), \alpha(1) = 3, 2, 1$, respectively. Therefore, we only need to investigate the influence of number of recurrent stages. In addition, we follow the original DETR and apply auxiliary losses to enhance the training of intermediate outputs of the glimpse-based decoders and apply LayerNorm [1] to help regularize the decoded glimpse representation $H_{dec}(i)$.

For each recurrent stage of the REGO, we use the decoder architecture of the original DETR for glimpse feature translation, but we do not use encoders and only use 2 decoding layers for a decoder. In decoders, the self-attention of encoders brings marginal benefits but consumes more computational resources, *e.g.* for REGO-DeformabelDETR-R50, adding the self-attention layers for all stages only improves AP, AP_{50}, and AP_{75} by 0.1, -0.1, 0.2, respectively, while introducing around 4 more GFLOPs and 9M more parameters. Without encoders, the complexity of the decoder in REGO is much smaller than the decoder used in the original DETR methods. The complexity analysis is presented in the experiment section. Besides the number of stages, we present other implementation details as follows. Firstly, we follow the default settings of the RoIAlign [17] and uses a 7 by 7 window for feature extraction. In addition, when extracting glimpse features, we attempt to use features from different levels of backbone in both multi-scale and single-scale DETR methods, but note that we do not use the FPN [22] to save costs. Also, the number of RoIs depends on the output of DETR and the number of stages. We will present more details in supplementary materials.

5. Experiment

5.1. Setup

We follow existing DETR methods [3] and perform evaluation using the MS COCO [24] dataset which has 118k training images and 5k validation images. We follow the MS COCO protocol and report the performance using the evaluation metrics of average precision (AP), AP at 0.5, AP at 0.75, and AP for small, medium, and large objects. The validation set is mainly used for evaluation.

We apply our method on the original DETR [3] and Deformable DETR [48] using their released codes. For training, we follow the original settings of the released codes for fair comparison, except that we also perform experiments with much fewer training epochs. For example, the original DETR detectors adopt 500 or 50 training epochs, while we mainly evaluate our method with 50 or 36 training epochs.

5.2. Performance Evaluation

In this section, we perform comprehensive comparison between the current DETR methods and our method. Ta-

Detectors	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	#Params (M)
FCOS [37]	R50	36	41.0	59.8	44.1	26.2	44.6	52.2	177	32
Faster RCNN - FPN [22]	R50	36	40.2	61.0	43.8	24.2	43.5	52.0	180	42
Faster RCNN - FPN [22]	R101	36	42.0	62.5	45.9	25.2	45.6	54.6	246	61
Mask RCNN [17]	X101	36	44.5	64.9	48.7	27.6	48.3	57.7	457	102
Cascade Mask RCNN [2, 5]	X101	36	46.6	65.1	50.6	29.3	50.5	60.1	627	135
TSP-RCNN [35]	R50	96	45.0	64.5	49.6	29.7	47.7	58.0	188	-
Efficient DETR [45]	R50	36	44.2	62.2	48.0	28.4	47.5	56.6	159	35
Sparse RCNN [34]	R50	36	44.5	63.4	48.2	26.9	47.2	59.5	-	-
DETR [3]	R50	500	42.0	62.4	44.2	20.5	45.8	61.1	86	41
DETR-DC5 [3]	R50	500	43.3	63.1	45.9	22.5	47.3	61.1	187	41
UP-DETR [10]	R50	300	42.8	63.0	45.3	20.8	47.1	61.7	86	41
Conditional DETR [28]	R50	50	40.9	61.8	43.3	20.8	44.6	59.2	90	44
Anchor DETR [41]	R50	50	44.2	64.7	47.5	24.7	48.2	60.6	151	-
SMCA [12]	R50	50	43.7	63.6	47.2	24.2	47.0	60.4	152	40
SMCA [12]	R101	50	44.4	65.2	48.0	24.3	48.5	61.0	218	58
DETR* [3, 28, 48] †	R50	50	39.3	60.3	41.4	18.5	42.4	57.5	88	44
DETR*-DC5 [3, 28, 48] †		50	41.3	62.8	43.6	21.0	44.5	59.4	189	
REGO-DETR* **(ours)**	R50	50	42.3	60.5	46.2	26.2	44.8	57.5	112	58
REGO-DETR*-DC5 **(ours)**		50	44.0	62.6	47.8	26.5	45.2	62.9	213	
Deformable DETR [48]	R50	36†	42.7	61.4	46.7	25.9	46.2	56.6	173	40
		50	43.8	62.6	47.7	26.4	47.1	58.0		
REGO-Deformable DETR **(ours)**	R50	36	44.8	63.8	48.7	27.0	48.0	60.2	190	54
		50	45.9	65.2	49.7	27.6	48.9	61.5		
Deformable DETR** [48]†	R50	50	46.4	65.3	50.6	30.0	49.8	61.4	173	40
	R101	50	47.2	66.6	51.1	28.5	50.9	62.4	240	59
	X101	50	47.7	67.2	51.4	29.3	51.2	62.8	417	105
REGO-Deformable DETR ** **(ours)**	R50	50	47.6	66.8	51.6	29.6	50.6	62.3	190	54
	R101	50	48.5	67.0	52.4	29.5	52.0	64.4	257	73
	X101	50	**49.1**	**67.5**	**53.1**	**30.0**	**52.6**	**65.0**	434	119

Table 1. Results of different detectors on the MS COCO *val* split. Baseline results are shaded. * Improve with 300 queries, reference points, and focal loss [28, 48]. ** Improve with iterative box refinement and two-stage processing. † Reproduced using released code.

ble 1 shows the overall results on MS COCO *val* dataset. In particular, we thoroughly investigate the performance of applying REGO on different DETR methods using different backbone networks and different training epochs.

Comparison with different DETR methods We have applied our proposed REGO on two major DETR detectors for evaluation. This include the vanilla DETR [3] method improved with 300 queries, reference points, and focal loss as described by [48] and the Deformable DETR [48]. We also presented the reported performance of RCNN-based methods [2, 22, 32, 34, 35, 37] and other DETR variants [10, 12, 28, 41, 45]. From the results in Table 1, we can observe that our method consistently improves different R50-based baseline methods by around 2 points in AP using 50 epochs. For example, using the original DETR, we boosts the performance from 39.3 AP to 42.3 AP at 50 training epochs. Moreover, by further cooperating with iterative box refinement and two-stage processing when using the Deformable DETR as baseline, the REGO helps improve the AP from 46.4 to 47.6 which is the highest score in all the compared DETR methods trained with 50 epochs and R50 backbone. This demonstrates that our proposed REGO is effective for improving DETR by gradually attending to more accurate object areas with RoIs.

Comparison at Fewer Training Epochs By applying the REGO on the Deformable DETR, we also compare the detection performance obtained at 36 training epochs used by many traditional detection methods [22, 37]. According to the Table 1, we help the Deformable DETR achieve 44.8 AP using 36 training epochs while the original Deformable DETR only achieves 42.7 with 36 epochs. Under the same training period, our method also helps surpass FPN and FCOS greatly. We also performed an extra experiment of REGO-DeformableDETR-X101 also trained with 36 epochs, obtaining 48.1, 67.4, 52.0 in AP, AP_{50}, and AP_{75}, respectively, which are higher than the CascadeR-CNN [2], proving that our REGO can reduce training costs for DETR effectively.

Comparison with Different Backbones We also investigate the effectiveness of REGO on different backbone networks, including R50 [18], R101 [18], and X101 [42]. Besides the improvements over R50 network, the results in Table 1 also show that REGO continues to improve the baseline DETR method with both R101 and X101 networks promisingly. In particular, with X101 backbone network, our Deformable DETR + REGO detector achieves the high-

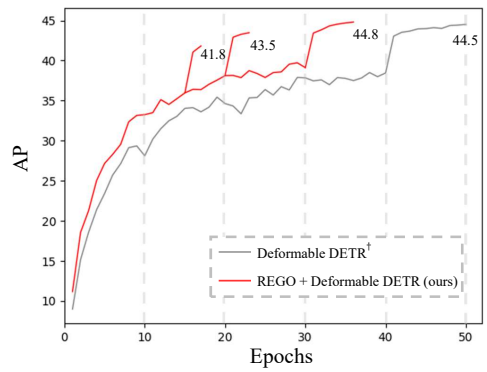

Figure 3. Convergence curves of Deformable DETR on the *val* set for whether using the proposed REGO. For REGO, we explore convergence performance by reducing the learning rate at the 15-th, 20-th, and 30-th epoch. †: Reproduced using released code.

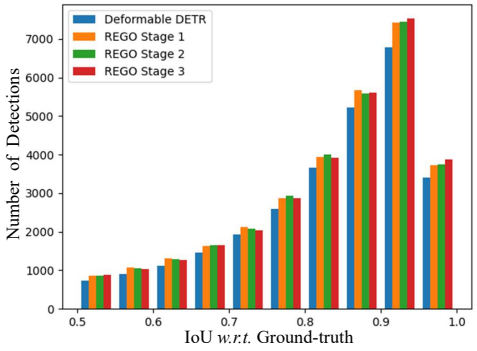

Figure 4. Histogram of correct detection results on the *val* set at different settings, *i.e.*, different IoU *w.r.t.* ground-truths and in different REGO stages. Note that the number of correct detection results of different stages share a similar amount ($\sim$ 30k boxes).

est AP among many state-of-the-art object detectors.

Convergence Analysis We further study the impact of REGO on actual convergence. Fig. 3 shows the detailed convergence curves of the Deformable DETR and the Deformable DETR with REGO. It shows that the REGO effectively speeds up the convergence and boosts the model performance promisingly comparing to the baseline. In particular, REGO helps achieve comparable performance with the baseline using only 30 epochs, *i.e.*, 40% less than the complete 50 training epochs used in the baseline. Comparing to the first DETR which requires 500 epochs, the REGO can help reduce about 94% of the total training period.

Complexity Analysis The extra computational complexity brought by the REGO is around 17 GFLOPs, which is only around 10% of the complexity of a Deformable DETR-R50 model while bringing around 28% acceleration in training (36 epochs *v.s.* 50 epochs). Furthermore, the complexity of our method remains the same when using larger and deeper backbone networks like R101 and X101 because of the identical implementation. The extra complexity *w.r.t.* these larger backbone network-based DETR are only around 7%, while the REGO brings more improvement rather than increasing depth of backbone networks. For example, the X101 improves R50 with 1.3 points in AP (46.4 to 47.7) for Deformable DETR at the cost of another 244 GFLOPs, while the REGO achieves similar results (47.6) with only extra 17 GFLOPs using the R50 backbone. Furthermore, we can also show in the supplementary materials that the Deformable DETR trained with REGO can already achieve around 1 point higher AP even without using REGO during inference, which means that the REGO directly helps original DETR learn better attention and offers detection improvement during inference *for free*.

Glimpse Stages	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Deformable DETR [48]	43.8	62.6	47.7	26.4	47.1	58.0
1 Stage (α = 1)	45.1	63.0	46.4	24.7	46.0	60.0
2 Stage (α = 2, 1)	45.6	65.1	49.3	27.4	48.7	61.2
3 Stage (α = 3, 2, 1)	45.9	65.2	49.7	27.6	48.9	61.5
4 Stage (α = 4, 3, 2, 1)	45.9	65.5	50.3	28.5	49.0	61.1

Table 2. Hyper-parameter study of the number of stages in REGO. Deformable DETR [48] is used as baseline.

Glimpse Scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1x	45.9	65.2	49.7	27.6	48.9	61.5
1.5x	45.8	65.0	50.1	27.6	48.9	61.3
2x	45.7	65.0	49.9	27.8	48.6	59.9

Table 3. Hyper-parameter study of the glimpse scale in REGO. REGO is implemented with 3 stages.

5.3. Ablation Study

Analysis of Different REGO Stages We first present the detection performance of applying different numbers of REGO stages in Table 2. The evaluated stages range from 1 to 4 where the glimpse scale in each stage varies accordingly as described in Section 4.3. We also present the baseline Deformable DETR result as reported in the paper [48]. The results show that the REGO with different numbers of stages improves the baseline performance greatly. A single processing stage can increase the mAP by more than 1 point. Applying more stages leads to further improvement. The performance of 3 and 4 stages is the best among compared settings. Although REGO with 4 stages achieves favorable performance, its improvement over the setting of 3 stages is marginal, implying that adding more than 3 processing stages in REGO could result in diminished benefits. We also disentangle the enlarging ratio from the 3-stage setting, *i.e.* making α=1,1,1, and only obtain 45.3 in mAP, showing that the glimpse design is useful.

We also study the quality of detection results from different stages in a 3-stage REGO module. A histogram chart of the numbers of correct detection results at different settings

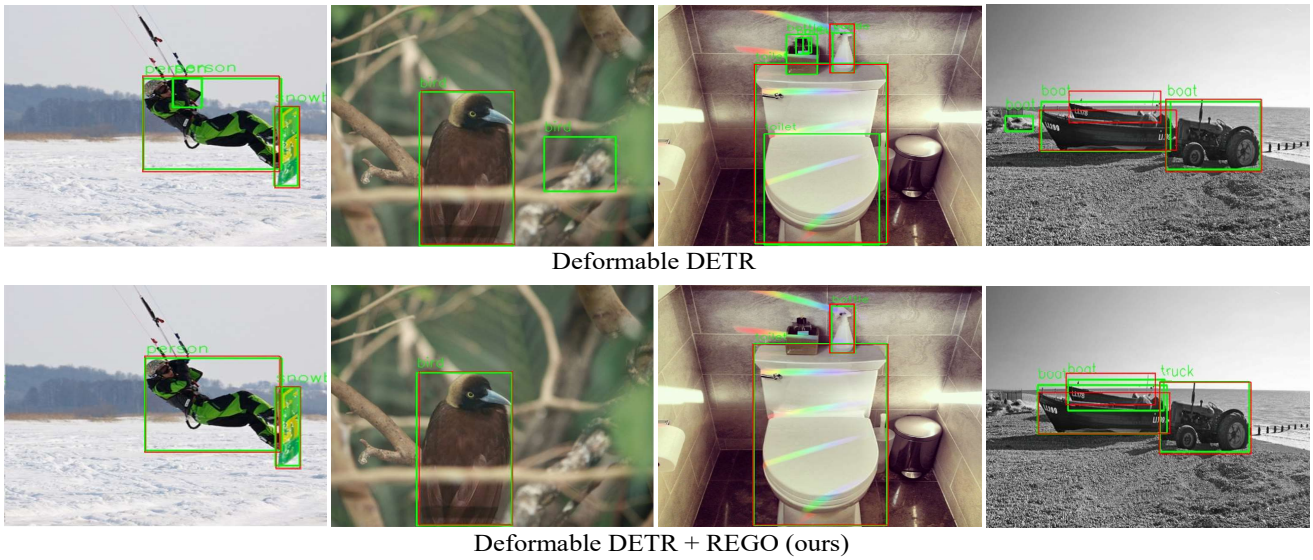

Figure 5. Visual detection results of the baseline Deformable DETR [48] and its variant with REGO. Green boxes are detection results while red boxes are ground-truths.

of Intersect-over-Union (IoU) *w.r.t.* ground-truths is shown in Fig. 4. A detection result is correct only if its IoU to a ground-truth is higher than 0.5 and its predicted label aligns with the ground-truth. In addition, the numbers of total detection results for different stages are similar, *i.e.*, around 30k boxes. Therefore, with the similar amount of correct detection results, the chart shows that all the REGO stages (red bars) help produce more accurate detection results than the baseline (blue bars) and their counterparts with fewer stages (yellow and green bars) for the right-most two groups of detection results. For example, the results of REGO with 3 stages contain more correct detection results whose IoU scores *w.r.t.* ground-truths are higher than 0.9. These results demonstrates that REGO with more stages continues to refine the detection by focusing on objects in the coarse-to-fine ROIs and learning better feature representations.

Analysis of Different Scales of Glimpse Area Table 3 shows the performance comparison of using different scales of glimpse area. The 1x, 1.5x, and 2x in the table represent the ratios of enlarging glimpse scales. For example, if using 2x and the default glimpse scales are 3.0, 2.0, 1.0 times larger than the previously detected bounding boxes, the actual glimpse scales are 6.0, 3.0, 2.0 times larger, respectively. We can find that the 1x setting already achieves the highest AP, and other settings achieve comparable but slightly lower AP. This suggests it is inappropriate to enlarge glimpse areas aggressively for implementing REGO.

5.4. Qualitative Results

We present some visual detection results to better illustrate the impact of REGO. We choose Deformable DETR [48] with R50 as baseline. Fig. 5 shows the results. Note that we choose the detection results whose confidence scores are higher than 0.5 for better visualization. From the figure, we can observe that the REGO indeed helps reduce both false positive and false negative results for the baseline method. Besides, the REGO can also help investigate the relations between different detected bounding boxes with the help of glimpse-based decoders. We will present some visual examples of the object relations learned with REGO in the supplementary materials.

6. Conclusion

We introduce a novel and effective technique, called REcurrent Glimpse-based decOder (REGO), to improve the Detection with Transformer (DETR) methods. By incorporating recurrent processing structure and learning glimpse features from coarse-to-fine RoIs, the REGO shows to both accelerate the convergence speed and boost the detection performance of different DETR methods consistently. We hope this study can contribute to future research on end-to-end and efficient detection methodologies.

Social Impacts and Limitations Our method can benefit various applications like self-driving. A potential limitation is that we still need several GPU days for training which is environmental costly. This can be mitigated by further improving the efficiency of both our REGO and the DETR.

Acknowledgement. Dr. Zhe Chen is supported by IH-180100002, and Dr. Jing Zhang is supported by ARC FL-170100117.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 2, 3, 6

[3] 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. 1, 2, 3, 5, 6

[4] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, 2022. 1

[5] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 6

[6] Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In *European conference on computer vision (ECCV)*, pages 71–86, 2018. 3

[7] Zhe Chen, Wanli Ouyang, Tongliang Liu, and Dacheng Tao. A shape transformation-based dataset augmentation framework for pedestrian detection. *International Journal of Computer Vision*, 129(4):1121–1138, 2021. 2

[8] Zhe Chen, Jing Zhang, and Dacheng Tao. Recursive context routing for object detection. *International Journal of Computer Vision*, 129(1):142–160, 2021. 2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *International Conference on Computer Vision*, pages 764–773, 2017. 2, 3

[10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Computer Vision and Pattern Recognition*, pages 1601–1610, 2021. 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[12] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. *arXiv preprint arXiv:2101.07448*, 2021. 2, 3, 6

[13] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *International Conference on Computer Vision*, pages 1134–1142, 2015. 3

[14] Spyros Gidaris and Nikos Komodakis. Attend refine repeat: Active box proposal generation via in-out localization. *arXiv preprint arXiv:1606.04446*, 2016. 3

[15] Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, pages 580–587, 2014. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision*, pages 2961–2969, 2017. 2, 5, 6

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 2, 6

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 1

[20] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European conference on computer vision (ECCV)*, pages 734–750, 2018. 2

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 1, 2, 5, 6

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2980–2988, 2017. 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 5

[25] Juhua Liu, Zhe Chen, Bo Du, and Dacheng Tao. Asts: A unified framework for arbitrary shape text spotting. *IEEE Transactions on Image Processing*, 29:5924–5936, 2020. 3

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 2

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[28] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *International Conference on Computer Vision*, pages 3651–3660, 2021. 2, 3, 6

[29] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014. 2

[30] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-

attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Computer Vision and Pattern Recognition*, pages 779–788, 2016. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 1, 2, 3, 6

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[34] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. SparseR-CNN: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 3, 6

[35] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *International Conference on Computer Vision*, pages 3611–3620, 2021. 3, 4, 6

[36] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020. 3

[37] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *International Conference on Computer Vision*, pages 9627–9636, 2019. 1, 6

[38] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 3

[40] Wen Wang, Yang Cao, Jing Zhang, and Dacheng Tao. Fp-detr: Detection transformer advanced by fully pre-training. In *International Conference on Learning Representations*, 2021. 2

[41] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 2, 3, 6

[42] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 6

[43] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[44] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Craft objects from images. In *Computer Vision and Pattern Recognition*, pages 6043–6051, 2016. 3

[45] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 3, 6

[46] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 1

[47] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022. 3

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 3, 5, 6, 7, 8