A APPENDIX

A.1 COMPLEXITY FOR DEFORMABLE ATTENTION

Supposes the number of query elements is N_q, in the deformable attention module (see Equation 2), the complexity for calculating the sampling coordinate offsets $\Delta \boldsymbol{p}_{mqk}$ and attention weights A_{mqk} is of $O(3N_qCMK)$. Given the sampling coordinate offsets and attention weights, the complexity of computing Equation 2 is $O(N_qC^2 + N_qKC^2 + 5N_qKC)$, where the factor of 5 in $5N_qKC$ is because of bilinear interpolation and the weighted sum in attention. On the other hand, we can also calculate $\boldsymbol{W}'_m \boldsymbol{x}$ before sampling, as it is independent to query, and the complexity of computing Equation 2 will become as $O(N_qC^2 + HWC^2 + 5N_qKC)$. So the overall complexity of deformable attention is $O(N_qC^2 + \min(HWC^2, N_qKC^2) + 5N_qKC + 3N_qCMK)$. In our experiments, $M = 8$, $K \leq 4$ and $C = 256$ by default, thus $5K + 3MK < C$ and the complexity is of $O(2N_qC^2 + \min(HWC^2, N_qKC^2))$.

A.2 CONSTRUCTING MULT-SCALE FEATURE MAPS FOR DEFORMABLE DETR

As discussed in Section 4.1 and illustrated in Figure 4, the input multi-scale feature maps of the encoder $\{\boldsymbol{x}^l\}_{l=1}^{L-1}$ ($L = 4$) are extracted from the output feature maps of stages C_3 through C_5 in ResNet (He et al., 2016) (transformed by a 1×1 convolution). The lowest resolution feature map $\boldsymbol{x}^L$ is obtained via a 3×3 stride 2 convolution on the final C_5 stage. Note that FPN (Lin et al., 2017a) is not used, because our proposed multi-scale deformable attention in itself can exchange information among multi-scale feature maps.

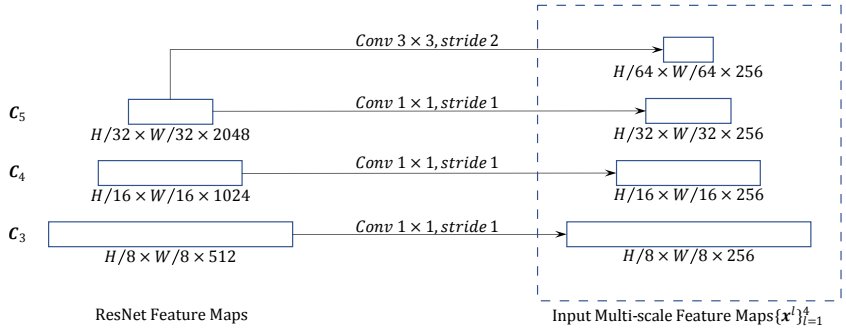


Figure 4: Constructing mult-scale feature maps for Deformable DETR.

A.3 BOUNDING BOX PREDICTION IN DEFORMABLE DETR

Since the multi-scale deformable attention module extracts image features around the reference point, we design the detection head to predict the bounding box as relative offsets w.r.t. the reference point to further reduce the optimization difficulty. The reference point is used as the initial guess of the box center. The detection head predicts the relative offsets w.r.t. the reference point $\hat{\boldsymbol{p}}_q = (\hat{p}_{qx}, \hat{p}_{qy})$, i.e., $\hat{\boldsymbol{b}}_q = \{\sigma\big(b_{qx} + \sigma^{-1}(\hat{p}_{qx})\big), \sigma\big(b_{qy} + \sigma^{-1}(\hat{p}_{qy})\big), \sigma(b_{qw}), \sigma(b_{qh})\}$, where $b_{q\{x,y,w,h\}} \in \mathbb{R}$ are predicted by the detection head. σ and σ^{-1} denote the sigmoid and the inverse sigmoid function, respectively. The usage of σ and σ^{-1} is to ensure $\hat{\boldsymbol{b}}$ is of normalized coordinates, as $\hat{\boldsymbol{b}}_q \in [0, 1]^4$. In this way, the learned decoder attention will have strong correlation with the predicted bounding boxes, which also accelerates the training convergence.

A.4 MORE IMPLEMENTATION DETAILS

Iterative Bounding Box Refinement. Here, each decoder layer refines the bounding boxes based on the predictions from the previous layer. Suppose there are D number of decoder layers (e.g., $D = 6$), given a normalized bounding box $\hat{\boldsymbol{b}}_q^{d-1}$ predicted by the $(d-1)$-th decoder layer, the d-th

decoder layer refines the box as

$$\hat{\boldsymbol{b}}_q^d = \{\sigma(\Delta b_{qx}^d + \sigma^{-1}(\hat{b}_{qx}^{d-1})), \sigma(\Delta b_{qy}^d + \sigma^{-1}(\hat{b}_{qy}^{d-1})), \sigma(\Delta b_{qw}^d + \sigma^{-1}(\hat{b}_{qw}^{d-1})), \sigma(\Delta b_{qh}^d + \sigma^{-1}(\hat{b}_{qh}^{d-1}))\},$$

where $d \in \{1, 2, ..., D\}$, $\Delta b_{q\{x,y,w,h\}}^d \in \mathbb{R}$ are predicted at the d-th decoder layer. Prediction heads for different decoder layers do not share parameters. The initial box is set as $\hat{b}_{qx}^0 = \hat{p}_{qx}$, $\hat{b}_{qy}^0 = \hat{p}_{qy}$, $\hat{b}_{qw}^0 = 0.1$, and $\hat{b}_{qh}^0 = 0.1$. The system is robust to the choice of b_{qw}^0 and b_{qh}^0. We tried setting them as 0.05, 0.1, 0.2, 0.5, and achieved similar performance. To stabilize training, similar to Teed & Deng (2020), the gradients only back propagate through $\Delta b_{q\{x,y,w,h\}}^d$, and are blocked at $\sigma^{-1}(\hat{b}_{q\{x,y,w,h\}}^{d-1})$.

In iterative bounding box refinement, for the d-th decoder layer, we sample key elements respective to the box $\hat{\boldsymbol{b}}_q^{d-1}$ predicted from the $(d-1)$-th decoder layer. For Equation 3 in the cross-attention module of the d-th decoder layer, $(\hat{b}_{qx}^{d-1}, \hat{b}_{qy}^{d-1})$ serves as the new reference point. The sampling offset $\Delta \boldsymbol{p}_{mlqk}$ is also modulated by the box size, as $(\Delta p_{mlqkx}\, \hat{b}_{qw}^{d-1}, \Delta p_{mlqky}\, \hat{b}_{qh}^{d-1})$. Such modifications make the sampling locations related to the center and size of previously predicted boxes.

Two-Stage Deformable DETR. In the first stage, given the output feature maps of the encoder, a detection head is applied to each pixel. The detection head is of a 3-layer FFN for bounding box regression, and a linear projection for bounding box binary classification (i.e., foreground and background), respectively. Let i index a pixel from feature level $l_i \in \{1, 2, ..., L\}$ with 2-d normalized coordinates $\hat{\boldsymbol{p}}_i = (\hat{p}_{ix}, \hat{p}_{iy}) \in [0, 1]^2$, its corresponding bounding box is predicted by

$$\hat{\boldsymbol{b}}_i = \{\sigma(\Delta b_{ix} + \sigma^{-1}(\hat{p}_{ix})), \sigma(\Delta b_{iy} + \sigma^{-1}(\hat{p}_{iy})), \sigma(\Delta b_{iw} + \sigma^{-1}(2^{l_i - 1}s)), \sigma(\Delta b_{ih} + \sigma^{-1}(2^{l_i - 1}s))\},$$

where the base object scale s is set as 0.05, $\Delta b_{i\{x,y,w,h\}} \in \mathbb{R}$ are predicted by the bounding box regression branch. The Hungarian loss in DETR is used for training the detection head.

Given the predicted bounding boxes in the first stage, top scoring bounding boxes are picked as region proposals. In the second stage, these region proposals are fed into the decoder as initial boxes for the *iterative bounding box refinement*, where the positional embeddings of object queries are set as positional embeddings of region proposal coordinates.

Initialization for Multi-scale Deformable Attention. In our experiments, the number of attention heads is set as $M = 8$. In multi-scale deformable attention modules, $\boldsymbol{W}_m' \in \mathbb{R}^{C_v \times C}$ and $\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$ are randomly initialized. Weight parameters of the linear projection for predicting A_{mlqk} and $\Delta \boldsymbol{p}_{mlqk}$ are initialized to zero. Bias parameters of the linear projection are initialized to make $A_{mlqk} = \frac{1}{LK}$ and $\{\Delta \boldsymbol{p}_{1lqk} = (-k, -k), \Delta \boldsymbol{p}_{2lqk} = (-k, 0), \Delta \boldsymbol{p}_{3lqk} = (-k, k), \Delta \boldsymbol{p}_{4lqk} = (0, -k), \Delta \boldsymbol{p}_{5lqk} = (0, k), \Delta \boldsymbol{p}_{6lqk} = (k, -k), \Delta \boldsymbol{p}_{7lqk} = (k, 0), \Delta \boldsymbol{p}_{8lqk} = (k, k)\}$ $(k \in \{1, 2, ...K\})$ at initialization.

For *iterative bounding box refinement*, the initialized bias parameters for $\Delta \boldsymbol{p}_{mlqk}$ prediction in the decoder are further multiplied with $\frac{1}{2K}$, so that all the sampling points at initialization are within the corresponding bounding boxes predicted from the previous decoder layer.

A.5 What Deformable DETR looks at?

For studying what Deformable DETR looks at to give final detection result, we draw the gradient norm of each item in final prediction (i.e., x/y coordinate of object center, width/height of object bounding box, category score of this object) with respect to each pixel in the image, as shown in Fig. 5. According to Taylor's theorem, the gradient norm can reflect how much the output would be changed relative to the perturbation of the pixel, thus it could show us which pixels the model mainly relys on for predicting each item.

The visualization indicates that Deformable DETR looks at extreme points of the object to determine its bounding box, which is similar to the observation in DETR (Carion et al., 2020). More concretely, Deformable DETR attends to left/right boundary of the object for x coordinate and width, and top/bottom boundary for y coordinate and height. Meanwhile, different to DETR (Carion et al., 2020), our Deformable DETR also looks at pixels inside the object for predicting its category.

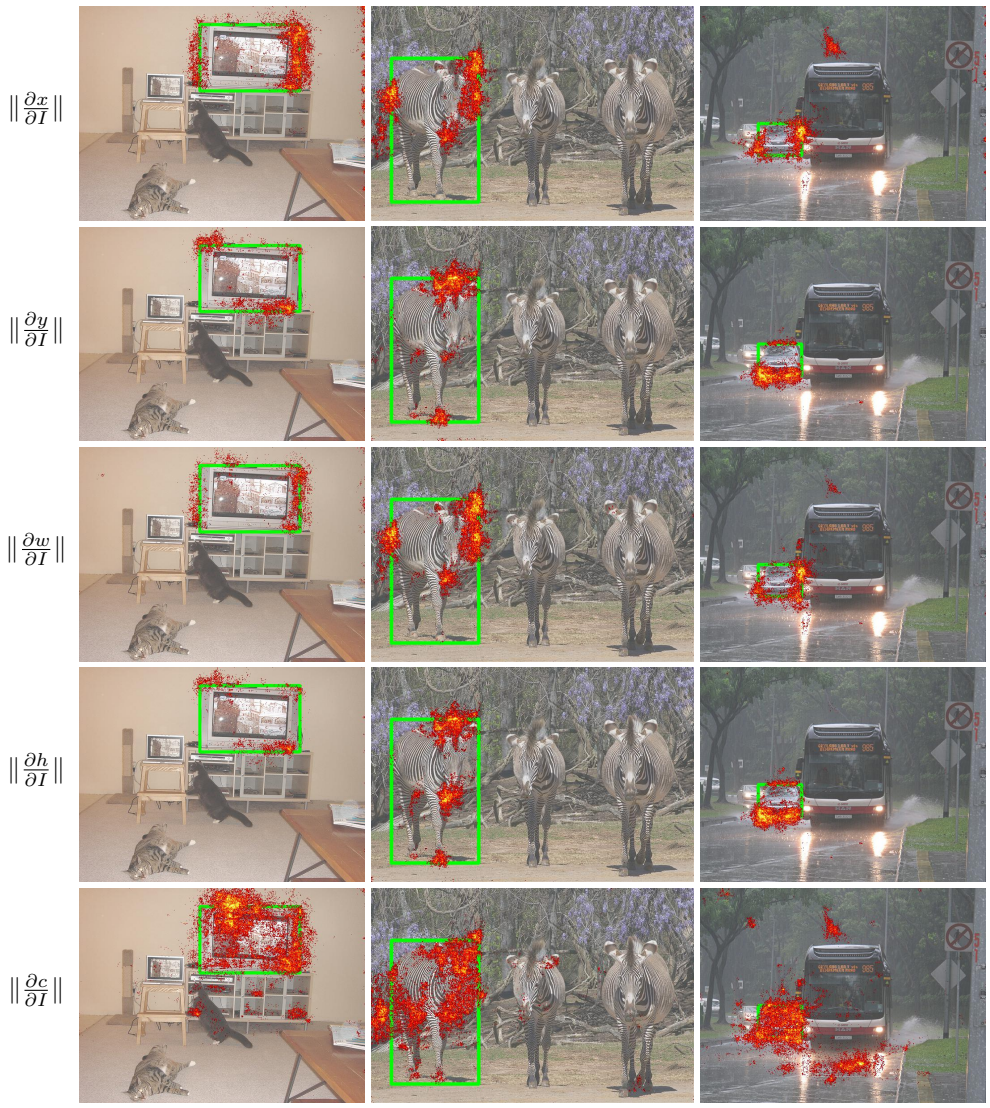
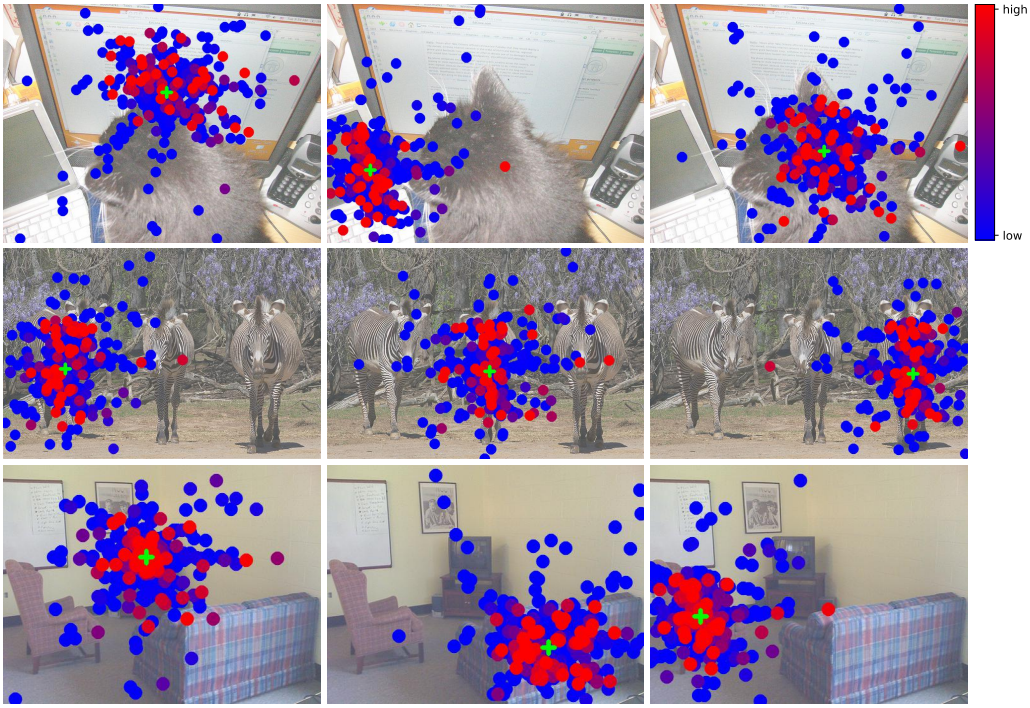


Figure 5: The gradient norm of each item (coordinate of object center (x, y), width/height of object bounding box w/h, category score c of this object) in final detection result with respect to each pixel in input image I.

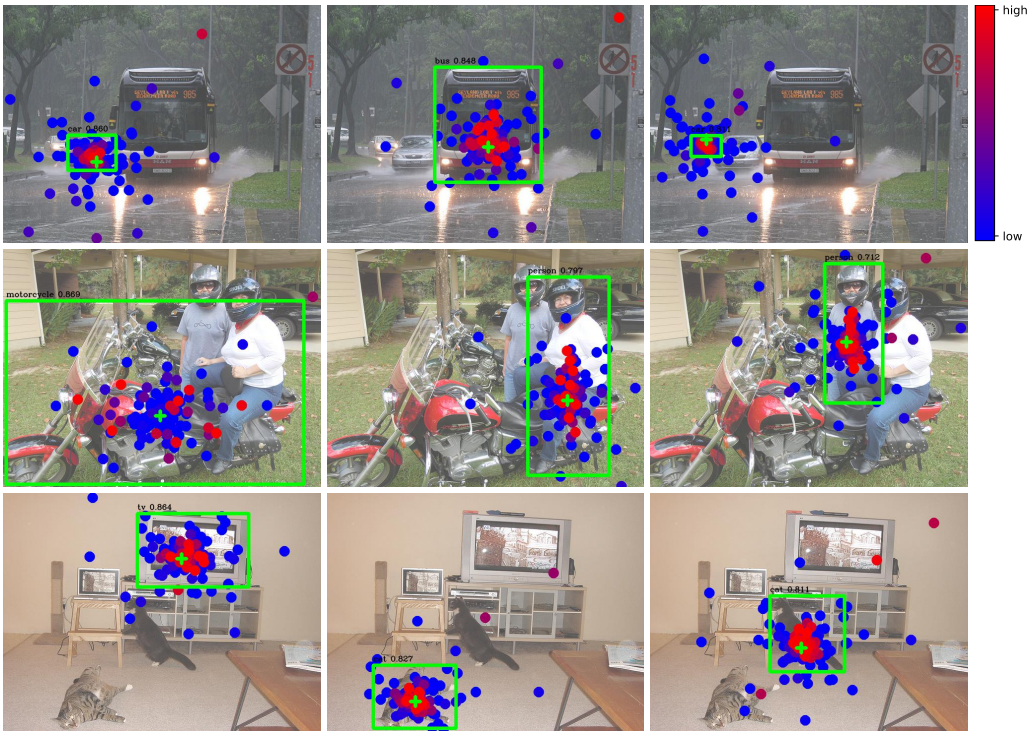
A.6 Visualization of Multi-scale Deformable Attention

For better understanding learned multi-scale deformable attention modules, we visualize sampling points and attention weights of the last layer in encoder and decoder, as shown in Fig. 6. For readibility, we combine the sampling points and attention weights from feature maps of different resolutions into one picture.

Similar to DETR (Carion et al., 2020), the instances are already separated in the encoder of Deformable DETR. While in the decoder, our model is focused on the whole foreground instance instead of only extreme points as observed in DETR (Carion et al., 2020). Combined with the visualization of $\|\frac{\partial c}{\partial I}\|$ in Fig. 5, we can guess the reason is that our Deformable DETR needs not only extreme points but also interior points to detemine object category. The visualization also demonstrates that the proposed multi-scale deformable attention module can adapt its sampling points and attention weights according to different scales and shapes of the foreground object.



(a) multi-scale deformable self-attention in encoder



(b) multi-scale deformable cross-attention in decoder

Figure 6: Visualization of multi-scale deformable attention. For readibility, we draw the sampling points and attention weights from feature maps of different resolutions in one picture. Each sampling point is marked as a filled circle whose color indicates its correspoinding attention weight. The reference point is shown as green cross marker, which is also equivalent to query point in encoder. In decoder, the predicted bounding box is shown as a green rectangle and the category and confidence score are texted just above it.

A.7 NOTATIONS

Table 4: Lookup table for notations in the paper.

Notation	Description
m	index for attention head
l	index for feature level of key element
q	index for query element
k	index for key element
N_q	number of query elements
N_k	number of key elements
M	number of attention heads
L	number of input feature levels
K	number of sampled keys in each feature level for each attention head
C	input feature dimension
C_v	feature dimension at each attention head
H	height of input feature map
W	width of input feature map
H^l	height of input feature map of l^{th} feature level
W^l	width of input feature map of l^{th} feature level
A_{mqk}	attention weight of q^{th} query to k^{th} key at m^{th} head
A_{mlqk}	attention weight of q^{th} query to k^{th} key in l^{th} feature level at m^{th} head
$\boldsymbol{z}_q$	input feature of q^{th} query
$\boldsymbol{p}_q$	2-d coordinate of reference point for q^{th} query
$\hat{\boldsymbol{p}}_q$	normalized 2-d coordinate of reference point for q^{th} query
$\boldsymbol{x}$	input feature map (input feature of key elements)
$\boldsymbol{x}_k$	input feature of k^{th} key
$\boldsymbol{x}^l$	input feature map of l^{th} feature level
$\Delta \boldsymbol{p}_{mqk}$	sampling offset of q^{th} query to k^{th} key at m^{th} head
$\Delta \boldsymbol{p}_{mlqk}$	sampling offset of q^{th} query to k^{th} key in l^{th} feature level at m^{th} head
$\boldsymbol{W}_m$	output projection matrix at m^{th} head
$\boldsymbol{U}_m$	input query projection matrix at m^{th} head
$\boldsymbol{V}_m$	input key projection matrix at m^{th} head
$\boldsymbol{W}'_m$	input value projection matrix at m^{th} head
$\phi_l(\hat{\boldsymbol{p}})$	unnormalized 2-d coordinate of $\hat{\boldsymbol{p}}$ in l^{th} feature level
$\exp$	exponential function
σ	sigmoid function
σ^{-1}	inverse sigmoid function