Deformable DETR:
Deformable Transformers for End-to-End Object Detection

Jifeng Dai
With Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li and Xiaogang Wang

SenseTime Research, USTC, CUHK
Previous Modern Object Detectors

• Rely on Hand-Crafted Components, e.g.,
  • anchor generation
  • rule-based training target assignment
  • non-maximum suppression (NMS) post-processing

• Not Fully End-to-End
  • complex combination of hand-crafted components
  • requiring manually adjustment (e.g., anchor size and NMS threshold) for specific datasets
DETR - The First End-to-End Object Detector

- Eliminate the need for hand-crafted components
- Achieve very competitive performance with previous modern object detectors

DETR - Issues

- Much longer training epochs to converge
  - 500 epochs on COCO, around 10 to 20 times slower than Faster R-CNN
- Low performance at detecting small objects

DETR - Issues

Both issues can be mainly attributed to the deficit of Transformer components in processing image feature maps

• Much longer training epochs to converge
  • 500 epochs on COCO, around 10 to 20 times slower than Faster R-CNN
• Low performance at detecting small objects

Revisit Multi-Head Attention in Transformers

• Enable neural networks to focus more on relevant elements of the input than on irrelevant parts.

Vaswani, Ashish, et al. "Attention is all you need." In NeurIPS 2017
Revisit Multi-Head Attention in Transformers

$$\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \right]$$

subjective to \( \sum_{k \in \Omega_k} A_{mqk} = 1 \)

Vaswani, Ashish, et al. "Attention is all you need." In NeurIPS 2017
Issues of Multi-Head Attention in Transformers

MultiHeadAttn\( (z_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \right] \)

- Long training schedules are required so that the attention weights can focus on specific keys
  - \( A_{mqk} \approx \frac{1}{N_k} \) at initialization, which leads to ambiguous gradients for inputs
    - \( N_k \) is the number of key elements
    - In the image domain, where the key elements are usually of image pixels, \( N_k \) can be very large and the convergence is tedious
- DETR requires much longer training epochs to converge
  - Attention modules processing image features are difficult to train

Vaswani, Ashish, et al. "Attention is all you need." In NeurIPS 2017
Issues of Multi-Head Attention in Transformers

\[
\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \right]
\]

- the computational and memory complexity can be very high
  - Computational complexity \( O(N_q C^2 + N_k C^2 + N_q N_k C) \),
    - \( N_q \) and \( N_k \) are the number of query and key elements,
    - \( C \) is the feature dimension
  - In the image domain, where the query and key elements are both of pixels,
    \( N_q = N_k \gg C \), the complexity is dominated by \( O(N_q N_k C) \)
- DETR delivers low performance at detecting small objects
  - Modern detectors use high-resolution feature maps to better detect small objects
  - high-resolution feature maps lead to unacceptable complexity

Vaswani, Ashish, et al. "Attention is all you need." In NeurIPS 2017
Issues of Multi-Head Attention in Transformers

\[
\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \right]
\]

- long training schedules are required so that the attention weights can focus on specific keys
- computational and memory complexity can be very high

The core issue is Transformer attention would look over all possible spatial locations

Vaswani, Ashish, et al. "Attention is all you need." In NeurIPS 2017
Efficient Sparse Attention in Image Domain

Dense attention (e.g., Transformer\cite{1}, Non-Local\cite{2})

look over all possible spatial locations

1D local attention (e.g., Image Transformer\cite{5})

look over pre-defined sparse spatial locations

2D local attention (e.g., Image Transformer\cite{5}, Stand-Alone\cite{6}, Local Relation\cite{7})

attention along each axis (e.g., Axial Attention\cite{3}, CCNet\cite{4})

much slower in implementation than traditional convolution with the same FLOPs

Deformable convolution is effective and efficient on image recognition. However, it lacks the element relation modeling mechanism.

Deformable DETR

- Deformable Attention

\[
\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right]
\]

- It only attends to a small set of key sampling points around a reference point, regardless of the spatial size of the feature maps.
- \( K \) is the total sampled key number (\( K \ll HW \))
Deformable DETR

• Deformable Attention

\[
\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right]
\]

• Equivalent to **Deformable Convolution**, when \( K = 1 \) and \( W'_m \) is fixed as an identity matrix

• Equivalent to **Transformer Attention**, when \( K = HW \) and the sampling points traverse all possible locations
Deformable DETR

• Deformable Attention

\[
\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right]
\]

• Multi-scale Deformable Attention

\[
\text{MSDeformAttn}(z_q, \hat{p}_q, \{x^l\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \left[ \sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\phi_l(\hat{p}_q) + \Delta p_{mlqk}) \right]
\]

- normalized coordinates $\hat{p}_q \in [0, 1]^2$ for the clarity of scale formulation
- function $\phi_l(\hat{p}_q)$ re-scales the normalized coordinates $\hat{p}_q$ to the input feature map of the $l$-th level
Deformable DETR

• Deformable Attention

\[
\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right]
\]

• Multi-scale Deformable Attention

\[
\text{MSDeformAttn}(z_q, \hat{p}_q, \{x^l\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \left[ \sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\phi_l(\hat{p}_q) + \Delta p_{mlqk}) \right]
\]

• In Transformer encoder, for each query pixel, the reference point \(\hat{p}_q\) is itself
• In Transformer decoder, the reference point \(\hat{p}_q\) is predicted from its object query embedding via a learnable linear projection followed by a sigmoid function
Deformable DETR
Experiments

Table 1: Comparison of Deformable DETR with DETR on COCO 2017 val set. DETR-DC5+ denotes DETR-DC5 with Focal Loss and 300 object queries.

<table>
<thead>
<tr>
<th>Method</th>
<th>Epochs</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
<th>APs</th>
<th>APm</th>
<th>AL</th>
<th>params</th>
<th>FLOPs</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN + FPN</td>
<td>109</td>
<td>42.0</td>
<td>62.1</td>
<td>45.5</td>
<td>26.6</td>
<td>45.4</td>
<td>53.4</td>
<td>42M</td>
<td>180G</td>
<td>26</td>
</tr>
<tr>
<td>DETR</td>
<td>500</td>
<td>42.0</td>
<td>62.4</td>
<td>44.2</td>
<td>20.5</td>
<td>45.8</td>
<td>61.1</td>
<td>41M</td>
<td>86G</td>
<td>28</td>
</tr>
<tr>
<td><strong>DETR-DC5</strong></td>
<td>500</td>
<td>43.3</td>
<td>63.1</td>
<td>45.9</td>
<td>22.5</td>
<td>47.3</td>
<td>61.1</td>
<td>41M</td>
<td>187G</td>
<td>12</td>
</tr>
<tr>
<td>DETR-DC5</td>
<td>50</td>
<td>35.3</td>
<td>55.7</td>
<td>36.8</td>
<td>15.2</td>
<td>37.5</td>
<td>53.6</td>
<td>41M</td>
<td>187G</td>
<td>12</td>
</tr>
<tr>
<td>DETR-DC5+</td>
<td>50</td>
<td>36.2</td>
<td>57.0</td>
<td>37.4</td>
<td>16.3</td>
<td>39.2</td>
<td>53.9</td>
<td>41M</td>
<td>187G</td>
<td>12</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>50</td>
<td>43.8</td>
<td>62.6</td>
<td>47.7</td>
<td>26.4</td>
<td>47.1</td>
<td>58.0</td>
<td>40M</td>
<td>173G</td>
<td>19</td>
</tr>
<tr>
<td>+ iterative bounding box refinement</td>
<td>50</td>
<td>45.4</td>
<td>64.7</td>
<td>49.0</td>
<td>26.8</td>
<td>48.3</td>
<td>61.7</td>
<td>40M</td>
<td>173G</td>
<td>19</td>
</tr>
<tr>
<td>++ two-stage Deformable DETR</td>
<td>50</td>
<td>46.2</td>
<td>65.2</td>
<td>50.0</td>
<td>28.8</td>
<td>49.2</td>
<td>61.7</td>
<td>40M</td>
<td>173G</td>
<td>19</td>
</tr>
</tbody>
</table>

Deformable DETR achieves better performance (especially on small objects) with 10× less training epochs.
Figure 2: Convergence curves of Deformable DETR and DETR-DC5 on COCO 2017 val set. For Deformable DETR, we explore different training schedules by varying the epochs at which the learning rate is reduced (where the AP score leaps).
Experiments

Using multi-scale inputs can effectively improve detection accuracy with 1.7% AP, especially on small objects with 2.9% AP_S.

Table 2: Ablations for deformable attention on COCO 2017 val set. “MS inputs” indicates using multi-scale inputs. “MS attention” indicates using multi-scale deformable attention. $K$ is the number of sampling points for each attention head on each feature level.

<table>
<thead>
<tr>
<th>MS inputs</th>
<th>MS attention</th>
<th>K</th>
<th>FPNs</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_S$</th>
<th>AP$_M$</th>
<th>AP$_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td>FPN (Lin et al., 2017a)</td>
<td>43.8</td>
<td>62.6</td>
<td>47.8</td>
<td>26.5</td>
<td>47.3</td>
<td>58.1</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td>BiFPN (Tan et al., 2020)</td>
<td>43.9</td>
<td>62.5</td>
<td>47.7</td>
<td>25.6</td>
<td>47.4</td>
<td>57.7</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>w/o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>w/o</td>
<td>41.4</td>
<td>60.9</td>
<td>44.9</td>
<td>24.1</td>
<td>44.6</td>
<td>56.1</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>4</td>
<td>w/o</td>
<td>42.3</td>
<td>61.4</td>
<td>46.0</td>
<td>24.8</td>
<td>45.1</td>
<td>56.3</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td>w/o</td>
<td>43.8</td>
<td>62.6</td>
<td>47.7</td>
<td>26.4</td>
<td>47.1</td>
<td>58.0</td>
</tr>
</tbody>
</table>
Experiments

Table 2: Ablations for deformable attention on COCO 2017 val set. “MS inputs” indicates using multi-scale inputs. “MS attention” indicates using multi-scale deformable attention. $K$ is the number of sampling points for each attention head on each feature level.

<table>
<thead>
<tr>
<th>MS inputs</th>
<th>MS attention</th>
<th>$K$</th>
<th>FPNs</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_S$</th>
<th>AP$_M$</th>
<th>AP$_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td>FPN (Lin et al., 2017a)</td>
<td>43.8</td>
<td>62.6</td>
<td>47.8</td>
<td>26.5</td>
<td>47.3</td>
<td>58.1</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td>BiFPN (Tan et al., 2020)</td>
<td>43.9</td>
<td>62.5</td>
<td>47.7</td>
<td>25.6</td>
<td>47.4</td>
<td>57.7</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>1</td>
<td></td>
<td>39.7</td>
<td>60.1</td>
<td>42.4</td>
<td>21.2</td>
<td>44.3</td>
<td>56.0</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>1</td>
<td>w/o</td>
<td>41.4</td>
<td>60.9</td>
<td>44.9</td>
<td>24.1</td>
<td>44.6</td>
<td>56.1</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>4</td>
<td></td>
<td>42.3</td>
<td>61.4</td>
<td>46.0</td>
<td>24.8</td>
<td>45.1</td>
<td>56.3</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>4</td>
<td></td>
<td>43.8</td>
<td>62.6</td>
<td>47.7</td>
<td>26.4</td>
<td>47.1</td>
<td>58.0</td>
</tr>
</tbody>
</table>

Increasing the number of sampling points $K$ can further improve $0.9\%$ AP
Experiments

Using multi-scale deformable attention, which allows information exchange among different scale levels, can bring additional 1.5% improvement in AP.
Because the cross-level feature exchange is already adopted, adding FPNs will not improve the performance.
## Experiments

Table 3: Comparison of Deformable DETR with state-of-the-art methods on COCO 2017 test-dev set. “TTA” indicates test-time augmentations including horizontal flip and multi-scale testing.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>TTA</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_S$</th>
<th>AP$_M$</th>
<th>AP$_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCOS (Tian et al., 2019)</td>
<td>ResNeXt-101</td>
<td></td>
<td>44.7</td>
<td>64.1</td>
<td>48.4</td>
<td>27.6</td>
<td>47.5</td>
<td>55.6</td>
</tr>
<tr>
<td>ATSS (Zhang et al., 2020)</td>
<td>ResNeXt-101 + DCN</td>
<td>✓</td>
<td>50.7</td>
<td>68.9</td>
<td>56.3</td>
<td>33.2</td>
<td>52.9</td>
<td>62.4</td>
</tr>
<tr>
<td>TSD (Song et al., 2020)</td>
<td>SENet154 + DCN</td>
<td>✓</td>
<td>51.2</td>
<td>71.9</td>
<td>56.0</td>
<td>33.8</td>
<td>54.8</td>
<td>64.2</td>
</tr>
<tr>
<td>EfficientDet-D7 (Tan et al., 2020)</td>
<td>EfficientNet-B6</td>
<td></td>
<td>52.2</td>
<td>71.4</td>
<td>56.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>ResNet-50</td>
<td></td>
<td>46.9</td>
<td>66.4</td>
<td>50.8</td>
<td>27.7</td>
<td>49.7</td>
<td>59.9</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>ResNet-101</td>
<td></td>
<td>48.7</td>
<td>68.1</td>
<td>52.9</td>
<td>29.1</td>
<td>51.5</td>
<td>62.0</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>ResNeXt-101</td>
<td></td>
<td>49.0</td>
<td>68.5</td>
<td>53.2</td>
<td>29.7</td>
<td>51.7</td>
<td>62.8</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>ResNeXt-101 + DCN</td>
<td></td>
<td>50.1</td>
<td>69.7</td>
<td>54.6</td>
<td>30.6</td>
<td>52.8</td>
<td>64.7</td>
</tr>
<tr>
<td>Deformable DETR</td>
<td>ResNeXt-101 + DCN</td>
<td>✓</td>
<td>52.3</td>
<td>71.9</td>
<td>58.1</td>
<td>34.4</td>
<td>54.4</td>
<td>65.6</td>
</tr>
</tbody>
</table>
Conclusion

• Deformable DETR is an end-to-end object detector, which is efficient and fast-converging.

• Compared with DETR, Deformable DETR can achieve better performance (especially on small objects) with 10× less training epochs.

• It enables us to explore more interesting and practical variants of end-to-end object detectors.

• We hope our work opens up new possibilities in exploring end-to-end object detection.

• Code is released at

  https://github.com/fundamentalvision/Deformable-DETR
We are hiring!

• Email: daijifeng@sensetime.com