MS-DETR: Efficient DETR Training with Mixed Supervision

Chuyang Zhao[12], Yifan Sun[1], Wenhao Wang[3], Qiang Chen[1], Errui Ding[1], Yi Yang[4], Jingdong Wang[1*]
[1] Baidu VIS [2] Beihang University [3] University of Technology Sydney [4] Zhejiang University
{zhaochuyang, sunyifan01, chenqiang13}@baidu.com
wangwenhao0716@gmail.com, yangyics@zju.edu.cn
{dingerrui, wangjingdong}@baidu.com

A. Experiments about the quality of candidates

We present a comparative analysis of candidate quality between our MS-DETR and the baseline model. We use Deformable DETR++ [1, 4] with 900 queries as our baseline and build our MS-DETR by applying mixed supervision to it. We compare the quality of candidates in terms of two metrics: the mean Intersection over Union (IoU) score and the count of candidates.

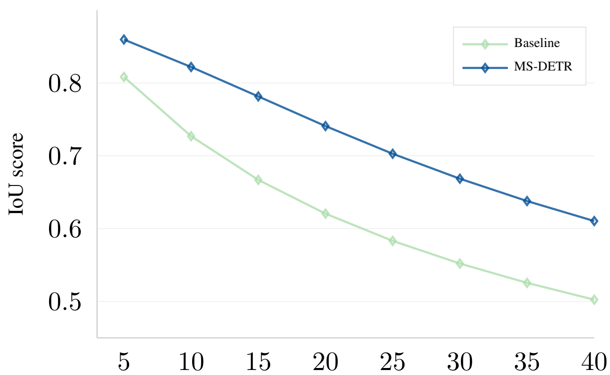

Figure 1. **Comparison of IoU scores of top-k candidates.** The x-axis corresponds to the value of k, and the y-axis corresponds to the averaged IoU scores of the top-k candidates of the COCO-2017 `val` set. One can see that the IoU scores of candidates in our MS-DETR surpass the baseline, which indicates the quality of the candidates is better with our approach.

We visualize the mean IoU score of top-k candidates in Figure 1. For each ground-truth object, we select queries with top-k IoU scores as its candidates. The mean IoU score is averaged over all ground-truth objects in the COCO-2017 [2] `val` set. We can see that with our mixed supervision, the mean IoU of top candidates surpasses the baseline by a large margin, indicating our MS-DETR generates better candidates.

In Figure 2, we visualize the count of high-quality candidates generated by the baseline and our MS-DETR, across varying IoU thresholds. High-quality candidates are queries with an IoU exceeding a specific threshold. One can see that with mixed supervision, our MS-DETR generates more high-quality candidates than the baseline.

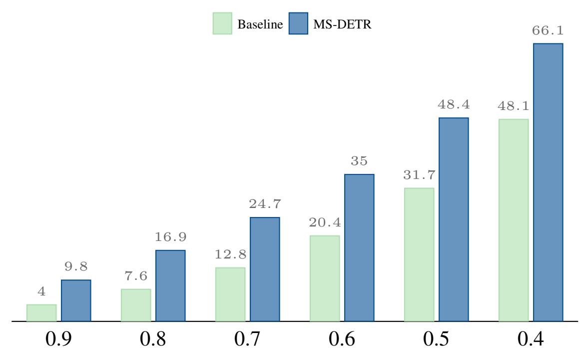

Figure 2. **Comparison of candidate numbers.** The x-axis corresponds to the value of IoU, and the y-axis corresponds to the number of candidates with IoU exceeding the specified threshold. The number of candidates is averaged across all ground-truth objects in the COCO-2017 `val` set. One can see that the number of high-quality candidates in our MS-DETR surpasses the baseline by a large margin.

B. Implementation details

We provide the implementation details of architecture (c) in Figure 2, which performs best among all MS-DETR variants.

The DETR decoder consists of multiple decoder layers. For clarity, we take one decoder layer for illustration. The input queries $\mathbf{Q}$ for each decoder layer first go through cross-attention layer to collect information from image features $\mathbf{I}$, resulting in the features after cross-attention:

$$\mathbf{Q}_{\text{ca}} = \texttt{CrossAttn}(\mathbf{Q}, \mathbf{I}), \tag{1}$$

The features after cross-attention layer are then fed into the self-attention layer, followed by a feed-forward network

*Corresponding author.

(FFN) to extract the features for one-to-one prediction:

$$\mathbf{Q}_{\text{sa}} = \texttt{SelfAttn}(\mathbf{Q}_{\text{ca}}), \quad \mathbf{Q}_{11} = \texttt{FFN}(\mathbf{Q}_{\text{sa}}), \tag{2}$$

The features after cross-attention layer are fed into an additional feed-forward network, yielding the features for one-to-many predictions:

$$\mathbf{Q}_{\text{1m}} = \texttt{FFN}(\mathbf{Q}_{\text{ca}}), \tag{3}$$

Both one-to-one and one-to-many predictions are derived using shared box and class predictors:

$$\begin{aligned} \mathbf{B}_{\text{1m}} &= \texttt{box}(\mathbf{Q}_{\text{1m}}), \quad \mathbf{S}_{\text{1m}} = \texttt{cls}(\mathbf{Q}_{\text{1m}}) \\ \mathbf{B}_{11} &= \texttt{box}(\mathbf{Q}_{11}), \quad \mathbf{S}_{11} = \texttt{cls}(\mathbf{Q}_{\text{1m}}), \end{aligned} \tag{4}$$

C. Details of one-to-many matching

We provide more details of our one-to-many matching introduced in Section 3.2. The algorithm establishes correspondences between the prediction sets $\{\mathbf{y}_i\}_{i=1}^{N}$ and the ground-truth object sets $\{\bar{\mathbf{y}}_i\}_{i=1}^{M}$, where N is the number of predictions, M is the number of ground-truth objects. Each element $\mathbf{y}$ in the prediction set consists of classification scores $\mathbf{s}$ and box prediction $\mathbf{b}$. Similarly, each element $\bar{\mathbf{y}}$ in the ground-truth object set consists of a ground-truth category $\bar{c}$ and bounding box $\bar{\mathbf{b}}$.

Following [3], we assign multiple predictions to one ground-truth object according to three criteria. We first compute the matching score between one prediction and ground-truth pair:

$$\texttt{MatchScore}(\mathbf{s}, \mathbf{b}, \bar{c}, \bar{\mathbf{b}}) = \alpha \cdot s_{\bar{c}} + (1-\alpha) \cdot \text{IoU}(\mathbf{b}, \bar{\mathbf{b}}).$$

We assign each prediction to the ground-truth object with the highest matching score to it. Then, we filter out low-quality queries with matching scores lower than the given threshold τ. Finally, for each ground-truth object, we select top-k predictions with highest matching scores as the matched results for this ground-truth object.

Optionally, we can merge the one-to-one matching set with our previously computed matching set as the final one-to-many matching set. This is because one-to-one matching results are derived using Hungarian matching, which may not align with our computed one-to-many matching results. For each matching item $(\mathbf{y}_{\sigma(i)}, \mathbf{y}_i)$ in the one-to-one matching set, if it does not exist in the one-to-many matching set, we add it to the one-to-many matching set. We empirically find this operation brings slightly ($0.1 \sim 0.2$ mAP) improvement. The detailed procedure is illustrated in Algorithm 1.

Algorithm 1 One-to-Many Matching Algorithm

Input: prediction set $\{\mathbf{y}_i\}_{i=1}^{N}$, ground-truth set $\{\bar{\mathbf{y}}_i\}_{i=1}^{M}$, threshold τ, top-k value k, score weight α, one-to-one matching set $L_{11} = \{(\mathbf{y}_{\sigma(i)}, \bar{\mathbf{y}}_i)\}_{i=1}^{N}$
Output: one-to-many matching set L_{1m}

function MATCHSCORE($\mathbf{s}$, $\mathbf{b}$, $\bar{c}$, $\bar{\mathbf{b}}$)
    return $\alpha \cdot s_{\bar{c}} + (1-\alpha) \cdot \text{IoU}(\mathbf{b}, \bar{\mathbf{b}})$
end function

Initialize L_{1m} as an empty set
for each ground-truth object $\bar{\mathbf{y}}_j = (\bar{c}_j, \bar{\mathbf{b}}_j)$ **do**
    Initialize an empty list L_j for top-k matches
    for each prediction $\mathbf{y}_i = (\mathbf{s}_i, \mathbf{b}_i)$ **do**
        score $\leftarrow$ MATCHSCORE($\mathbf{s}_i, \mathbf{b}_i, \bar{c}_j, \bar{\mathbf{b}}_j$)
        if *score* $> \tau$ **then**
            Add $(\mathbf{s}_i, \mathbf{b}_i, \textit{score})$ to L_j
        end if
    end for
    Sort L_j by *score* in descending order
    Keep the top-k elements of L_j
    Add elements from L_j to L_{1m}
end for

for each pair $(\mathbf{y}_{\sigma(i)}, \mathbf{y}_i)$ in L_{11} **do**
    if $\mathbf{y}_i \neq \varnothing$ **and** $(\mathbf{y}_{\sigma(i)}, \mathbf{y}_i) \notin L_{\text{1m}}$ **then**
        Append $(\mathbf{y}_{\sigma(i)}, \mathbf{y}_i)$ to L_{1m}
    end if
end for

return L_{1m}

References

[1] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022. 1

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 1

[3] Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, and Philipp Krähenbühl. Nms strikes back. *arXiv preprint arXiv:2212.06137*, 2022. 2

[4] 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. 1