DETR with multi-level decoder

Yuan Ma

March 2024

1 Introduction

In the past few years, object detection has evolved from traditional methods based on complex hand-crafted features to more advanced deep learning detector. However, popular deep learning object detection frameworks such as R-CNN [3] and its variants [5, 6, 17], still require complex multi-stage pipelines which heavily rely on handcraft features. Detection Transformer [2], known as DETR, emerged as a revolutionary approach that simplified the detection pipeline by leveraging the transformer architecture, which is traditionally used in natural language processing. Unlike other detection models, DETR enables end-to-end object detection without complex hand-designed components. This new approach quickly gained attention in the research community due to its elegance and efficiency.

However, the baseline DETR is notorious for its low accuracy on small objects and unacceptable long training schedule. Additionally, it is computational expensive for DETR to process high-resolution images, presenting hurdles in practical applications. In response to these challenges, Deformable DETR [12] has introduced deformable multi-head attention. This adaptation enhanced the model's ability to process high-resolution images due to its efficiency thus improving the detection accuracy of small object. Moreover, it proposed multi-scale deformable attention that extend attention mechanism to multi-scale thanks to its efficiency. Although such multi-scale design achieves state-of-the-art benchmarks later in DINO[22], for multi-scale features, the query length in transformer encoder is still too long to be computed efficiently. Therefore researchers have started to investigate towards efficient usage of multi-scale features in DETR. For example, Lite DETR[10] proposed interleaved update and iterative high-level feature cross-scale fusion. It frequently updates higher-level features with attention, but only computes the lower-level features in the last encoder layer to achieve an efficient feature fusion. And RT-DETR[16] only performs attention on the highest level of features instead of conducting comprehensive interaction between all the multi-scale feature maps. Feature fusion is then conducted among raw outputs from backbone and the feature from transformer encoder. In this way, it decreases the computation cost dramatically compared to the multi-scale structure proposed by deformable DETR. However, all these works focus on multi-scale feautre fusion at the transformer encoder, the way object queries interact with multi-scale features with deformable attention at the transformer decoder has not been investigated.

On the other hand, a series of works focus on optimizing object queries[3, 14, 17, 20, 21]. Object queries refer to tokens at DETR decoder side, which interfere with image features to predict position of the object. Anchor DETR[20] argues that object queries are not able to focus on a specific region, causing slow convergence during training. So it aligns anchor points to object queries to force object query focusing on a small area to accelerate model convergence. DETR with dynamic anchor box(DAB-DETR)[13] describes the role of object query as detecting similarity between keys and queries. Similar to anchor DETR, DAB-DETR find object query tends to be multi-pattern so that it is hard to optimize. But instead of setting prior points as anchors, it argues queries with a 4 dimension prior anchor box work better for DETR. Such anchor box design allows DETR to dynamically update the anchor box layer by layer, achieving both faster convergence and better accuracy. While the prior information for object query solve the multi-pattern issue, few may notice that the object queries interact with the same feature map in different decoder layers is also problematic: It introduces redundant information, causing a low efficiency and preventing decoder to get optimized detection accuracy. As illustrated in Fig.1, the accuracy of the original DETR only occurs minimum accuracy drop when cutting the last two decoder layers. And most of the performance comes from the first three layers. The same trend can also be found at DAB-DETR even with the upgraded object queries.

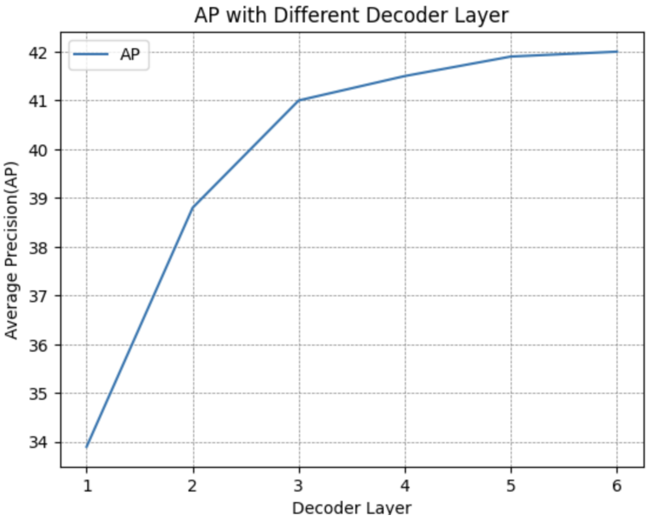
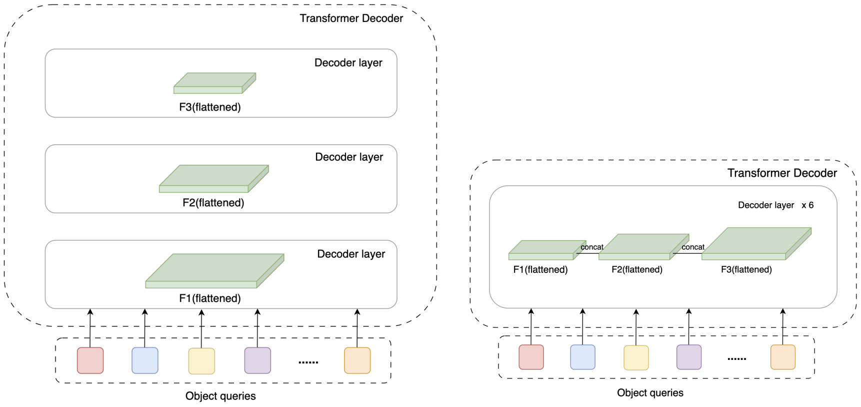


Figure 1: Average precison of DETR with different decoder layer

Motivated by this observation, we propose the DETR with Multi-level Decoder

(MLD-DETR). The proposed multi-level design leverages the abundant information from multi-scale features, in the meanwhile it naturally aligns with the function of dynamzic anchor box proposed in DAB-DETR. Intuitively, to dynamically locate an object, one must first form an initial hypothesis based on overview of the image, and locating the object by identification of specific details. In other words, we are detecting the object with the help of different semantic information. Based on this idea, we introduce a multi-level decoder where each layer of the object queries are paired with feature maps containing different semantic representations, enabling more efficient information retrieval and dynamic query updates. To achieve this goal, our proposed multi-level decoder vertically sets the multi-scale feature maps instead of a parallel one like other works such as deformable DETR and RT-DETR (as shown in Fig.2.(a) and Fig.2.(b), respectively). We use an adaptive Feature Pyramid Network to generate maps with different semantic information, which will be discussed in the proposed method section for more details. Note that although we take multi-scale features from backbone, they are resized to single-scale ones. Thus we conclude our model as a single-scale detector which leverage abundunt semantic information from multi-scale features. Our proposed method is simple and effective: with minimum changes, we observed a 1.4% mean average precision increase on COCO2017 dataset compared to our baseline model (DN-DETR), with almost no extra computational cost.



(a) Ours multi-scale design (b) Multi-scale structure in DETR family

Figure 2: Comparision between our proposed multi-scale method and traditional multi-scale method in DETR

In summary, we describe our contributions as follows: 1. we arrange various feature maps to object queries in different DETR transformer decoder layers, solving the inefficiency issues. 2. we propose a novel multi-level DETR pipeline with a unique vertical arrangement of feature maps, offering an alternative methodology in the development of DETR models that involves multi-scale features. 3.

With proposed DETR with Multi-Level Decoder(MLD-DETR), our model can achieve better accuracy as well as faster convergence with almost zero additional computation cost compared to the baseline.

2 Related Works

Prior information for object query DETR is notorious for its slow convergence and low accuracy on small objects. Lots of researchers believe it is because of the poor behavior of object query during training. So numerous works have been proposed to modify DETR's query design[3, 14, 17, 20, 21], aiming for faster training and better performance. For example, DETR-SMCA[3] proposed a spatial-aware cross-attention to accelerate the convergence of training. Anchor DETR[20], provides anchor points as prior information for object queries. DAB-DETR is one of the representation works of query design, it reveals that the slow convergence of DETR is due to the multi-pattern nature of object queries in DETR. To address this, the proposed DAB-DETR (Dynamic Anchor Box DETR) introduces a novel approach by using 4D anchor boxes (x, y, w, h) as prior information to force queries to learn a detection pattern for a single object, then updated them layer-by-layer. This method provides better spatial priors, considering both the position and size of each box, deepening the understanding of queries in DETR. It not only accelerates DETR's training but also adapts the positional prior to match objects of varying scales, noting a significant advancement over previous works. DN-DETR[11] is an enhanced version of DAB-DETR, it accelerates the training phase via providing model noised boxes to help object query better capture the position of objects. It is one of the best single scale DETR and the denoising process does not change the structure of DAB-DETR, so we take DN-DETR as our baseline model.

Multi-scale structures for DETR Deformable DETR represents another significant evolution in Detection Transformer, addressing primary limitations of the standard DETR model and introducing a multi-scale structure to DETR family. It modifies the standard DETR model by introducing deformable attention modules. These modules enable the model to focus attention on a set of key sampling points around the reference points, rather than the entire image. Such selective attention mechanism is particularly beneficial for detecting small objects. More importantly, it reduces the computation cost of multi-head attention, so that the deformable attention can extended to multi-scale deformable attention. Thus DETR can achieve multi-scale based object detection, which is a huge enhancement over single-scale-based one. Although it still requires a huge amount of computational resources given the unacceptable token length, the outstanding accuracy makes it a baseline for multi-scale DETR presented later. Recently, although the state-of-the-art real-time object detection benchmark RT-DETR proposed a more efficient encoder, multi-scale design in the transformer decoder also follows the same design with Deformable DETR.

RT-DETR achieves state-of-the-art performance on real-time object detection.

It beats yolo family which have dominated the real time detection for years, demonstrating the great potential of DETR. To achieve a high accuracy with little extra computational cost, how to use multi-scale features efficiently becomes a central topic. In this paper, author did very elaborate experiments on multi-scale features which finds out features fusion appears in Deformable DETR is not optimal and the FPN like structure[4, 9, 12] tend to have better accuracy with lower computational cost. Consequently, they proposed the attention-based Intra-scale Feature Interaction (AIFI) module for high level features and the CNN-based Cross-scale Feature-fusion Module (CCFM) for other features which achieves an optimal multi-scale structure for DETR. Along with another queries selected method, the RT-DETR achieves state-of-the-art benchmarks and sets up a standard for multi-scale DETR. Our work, although arranging multi-scale features in different manner and downsampling them into same size, does not conflict with RT-DETR's standard, instead we argue that our proposed model provides other possibilities for DETR that involves multi-scale features.

3 Proposed Method

3.1 Overview of Multi-scale features

Multi-scale feature is an ideal candidate to feed into the transformer decoder to solve the inefficient issue. It has been widely regarded as a valuable component for vision tasks[8] since 1960. The traditional multi-scale structure approach processes visual information at various resolutions, mimicking the human visual system's capability to perceive fine details when focusing closely on an object and broader shapes from a distance.

For modern deep neural networks, there is a deep representation of images due to the substantial depth of the network. The size of the feature maps decreases as the depth of the network increases. Feature Pyramid Network (FPN)[12] defined the **network stage** as layers that produce output maps of the same size and denoted the feature maps of the last three stages of ResNet50 as C3, C4 and C5; we follow the same terms in our work for clarity. Researchers believe that features from different network stages contain not only varying spatial resolutions but also diverse semantic information: features from earlier stages tend to have lower-level semantic information, like edges or corners, while deeper-stage features are rich in semantic information, such as shapes and patterns. Deep features provide superior information for object detection, but it is computational expensive to generate deep multi-scale representations in parallel. FPN solves this issue by efficiently merging features through a simple lateral addition and feature projection, achieving better detection accuracy with little computational cost. Inspired by this design, our work follows a similar pipeline, while FPN is aimed to generate multi-scale features that contains the deep semantic information, leveraging the various semantic information at different stages is what we intend to achieve.

3.2 MLD-DETR

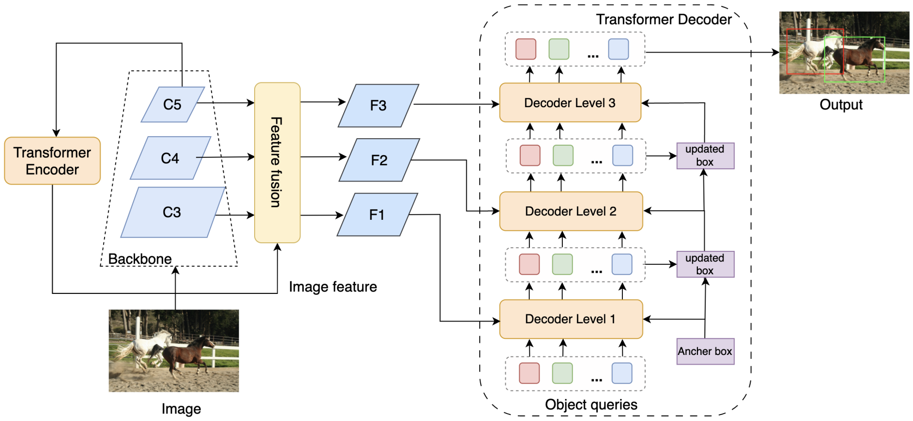


Figure 3: Overall architecture of our proposed decoder. The image feature from transformer encoder is passing to the bottom of feature fusion module, which is fused with C3. And then the generated feature F1, F2, F3 are feeding into different layers of transformer decoder.

Feature fusion module The overall architecture of the proposed work is shown in Fig.3. Our work fuses features via a feature fusion module and then these fused features are sequentially fed into the transformer decoder. To achieve simplicity and efficiency, we borrow basic ideas from the FPN, fusing features with element-wise addition and lateral connections. However, our baseline model has substantial differences compared to FPN: 1. Instead of interpolating high-level feature maps to a larger spatial size, we employ a straightforward maxpooling operation to downsample the low-level feature map, resizing the feature maps to make it feasible to perform element-wise additions across different feature maps. Thus the proposed work is technically a single-scale Detector. In contrast to the traditional FPN, our work fuses features in the opposite order to generate features, as illustrated in Fig.4. From our analysis experiments, such structure can achieve optimal performance. Specifically, we use the image feature generated by the transformer encoder, which is the semantically strongest representation, as the base feature. Note that this image feature has been flattened into 3-dimension which is able to be processed by transformer, but we still measure its spatial shape as $\frac{H}{16}$ and weight $\frac{W}{16}$ for convenience, in which H and W are original size of image. We add the base image feature with reshaped $C3$ (denoted as $C3'$) to generate an enhanced feature map $F1$: the reshaping operation consists of a maxpooling operation to downsample spatial resolution of $C3$ ($\frac{H}{4}$, $\frac{W}{4}$) by a factor of 4 and a linear projection layer to align the number of channels in C3 with our base map. Unlike the FPN, we do not perform another 3×3 convolution on the enhanced feature because we intend to remain the features from transformer unchanged. $F2$ and $F3$ are generated following the same manner. We concludes the whole process as follows:

$$F_1 = x + \text{Conv}_{1\times 1}(\text{maxpooling}(c_3)), \tag{1}$$
$$F_2 = F_1 + \text{Conv}_{1\times 1}(\text{maxpooling}(c_4)), \tag{2}$$
$$F_3 = F_2 + \text{Conv}_{1\times 1}(c_5) \tag{3}$$

where x represents the image features generated by the transformer encoder. Note that We do not need to reshape $C5$ because it has the same size as the image feature and all the feature maps from the backbone will be flattened during the fusion process. In this way, we create multi-level features that contain different semantic information which will be used in the different decoder layers.

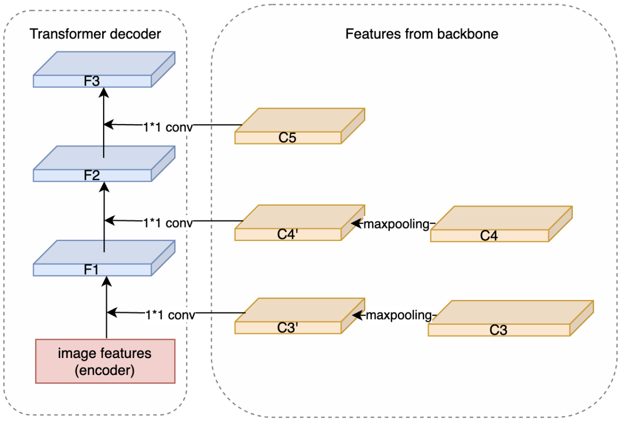


Figure 4: Multi-scale fusion module. Image features are generated by transformer encoder. Note that we downsample C4 and C3 into C4' and C3' respectively, which have the same size as image features.

Proposed transformer decoder For the naive detection transformer decoder, there are six identical decoder layers. In this work, we evenly arrange six decoder layers into three levels, feeding $F1$, $F2$ and $F3$ generated by our customized feature fusion module into three different decoder levels. Feature $F1$ is enhanced by shallow-stage feature $C3$, and we feed it into first level of the decoder (layer 1 and layer 2). Then we feed F2 and $F3$ successively into the next two decoder levels as they contain semantically stronger information. Unlike any previous works that concatenate multi-scale features and feed them into decoder layer simultaneously[12, 16, 22, 24], we set them in cascade in decoder. In this way, we can add abundant semantic information into different decoder levels, fulfill the role of dynamic anchor box, and solve the inefficiency issue we discussed before. The query design is the same as the original DAB-DETR: it provides four-dimensional prior information to object queries and then updates it layer by layer. Moreover, it also modulates the positional attention maps by dividing the relative anchor width and height from its x and y part separately to match with objects of different scales:

$$\text{ModulateAttn}((x,y),(x_{\text{ref}},y_{\text{ref}})) = \left(PE(x)\cdot PE(x_{\text{ref}})\frac{w_{q,\text{ref}}}{w_q} + PE(y)\cdot PE(y_{\text{ref}})\frac{h_{q,\text{ref}}}{h_q}\right)/\sqrt{D}, \tag{4}$$

where $(x,\ y)$ and $(x_{ref},\ y_{ref})$ are coordinate of two points, w_q and h_q are the width and height of the anchor A_q, and D is the scaling factor. PE denotes position encoding which is discribed in [2]. $w_{q,ref}$ and $h_{q,ref}$ are the reference width and height that are determined by a shared MLP at different decoder layers:

$$w_{q,ref}, h_{q,ref} = \sigma(\text{MLP}(C_q)), \tag{5}$$

where C_q is content embedding (object query) and activation function σ is chosen as Relu in this paper. Because in our proposed structure object queries interact with different feature maps, we implement a layer-aware ModulateAttn that learns $w_{q,ref}$ and $h_{q,ref}$ by independent MLP in different levels rather than a shared one. However, we only observe a slightly increase on accuracy compared to the baseline which uses only one MLP, demonstrating that the behaviors of object queries are highly related at different levels even though the feature maps are different from each other. Thus in our work we choose to use a shared MLP in different levels for efficiency.

4 Experiments

4.1 Dataset and model parameters

Following the common practice with other state-of-the-art DETR models, we trained our model on COCO train2017 dataset and evaluate it on COCO val2017. We use Mini COCO for ablation experiments. Model parameters settings are almost the same as DN-DAB-DETR for consistency as it serves as our baseline model.

4.2 Training settings and model comparison

The experiments results are shown in Table.1. Our training process is also aligned with most of DETR settings for fair comparison. We use AdamW as our optimizer with weight decay 10^{-4} and set learning rate as 1×10^{-4} to train for 50 epochs, which drops by 0.1 after 40 epochs. The number of object query is selected as 300. The proposed model is trained on a single node with 2 Nvidia Titan X GPUs, and the batch size for each GPU is 2 for memory consideration. Due to hardware limitations, we replicated DN-DAB-DETR with corresponding training settings and hyperparameters for fair comparison. In Table.1, we use $*$ to indicate the replicated experiments. Note that although our model incorporates multi-scale features, it is technically a single-scale model that does not include any large-size feature maps. Therefore, the comparison between our proposed model and the single-scale DETR is valid. Furthermore, we also compare our model with other single scale state-of-the-art DETR which trained on better hardware. Also note that all the backbone of DETRs listed for comparison are ResNet50. Comparing to DN-DETR, the detection accuracy of our proposed model increases by around 1.4% (from 41.9% to 43.3%) with little extra computational cost. Moreover, we also observe a faster convergence in the early stage. Our proposed model can achieve similar accuracy with only 30 epochs (41.9%) compared to the our baseline model. Furthermore, our model outperform other state-of-the-art DETR such as Anchor DETR, Conditional DETR, which requires more advanced hardware such as Nividia A100 and V100.

Model	epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	params
DN-DAB-DETR*	50	41.9	61.9	44.7	60.0	45.6	21.5	44 M
MLD-DETR (ours)	50	43.3	63.7	46.0	62.0	46.7	23.1	44 M
MLD-DETR	30	41.9	62.1	44.6	60.6	45.2	21.8	44 M
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41 M
Anchor-DETR	50	42.1	63.1	44.9	22.3	46.2	60.0	39 M
DAB-DETR	50	42.2	63.1	44.7	21.5	45.7	60.3	44 M
Conditional DETR	50	40.9	61.8	44.3	20.8	44.6	59.2	44 M

Table 1: Performance comparison of different DETR. The $*$ indicates our replicated experiments. Note that the first three model are trained with our local machine. All models are

4.3 Detection accuracy of different decoder layers

In this subsection, we conduct ablation experiments of different decoder layers to demonstrate that the proposed method also solve the inefficiency issue in the DETR transformer decoder. The experiment itself is straight-forward: In DETR, the final prediction is computed by a 3 layers Feed Forward Network(FFN). Additionally, the prediction loss is computed at each transformer decoder layer with the shared FFN for better performance. Thus we just evaluate the results with output from different layers on COCO eval to observe the detection accuracy of different layers. We mainly compare the accuracy increase with respect to last two layers because they are responsible for most of the inefficiency issue in other DETR models. And we do not take last one layers into consideration as there is no additional features introduced. Fig.6 illustrates the experiment result: for original DETR, the accuracy increase of last two layers is only 0.5%, while the last two layers of our proposed structure contribute 2% detection accuracy, which is four times better than original DETR. Thus, we can conclude that our multi-level decoder design solve the inefficiency issues appears in other DETR transformer decoder.

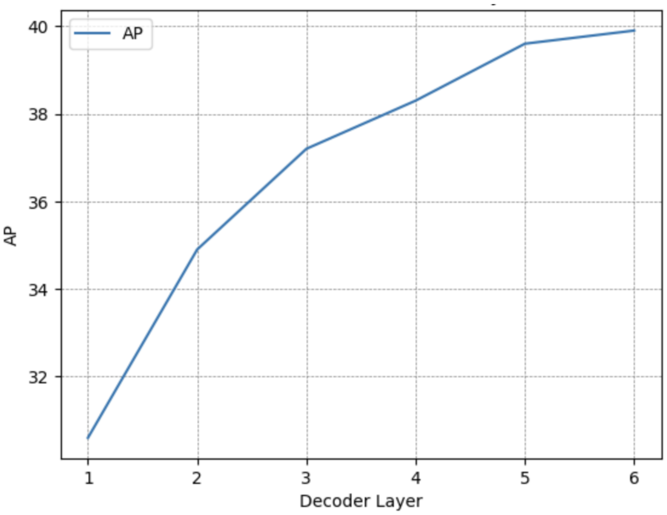


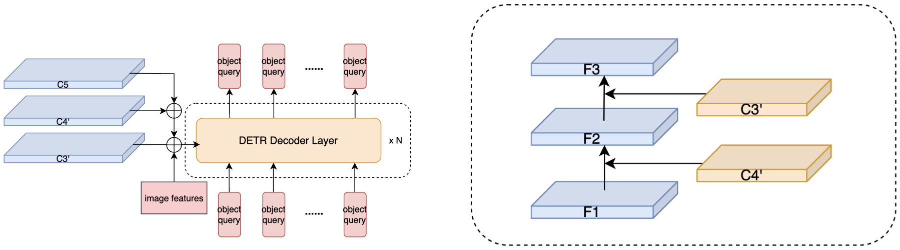
Figure 5: MLD-DETR's detection accruacy of different decoder layers

4.4 Ablation Study on Mini COCO

Overview Although we have observed that proposed model demonstrates significant improvement over the baseline model and significantly inprovement the efficiency of the DETR transformer decoder, there are several questions still required to be solved: 1. The improvements may not come solely from multi-level features and multi-level design; the additional information provided by the backbone itself could be the reason for the accuracy improvement. 2. The way we fuse feature maps is not aligned with the principle of the Feature Pyramid Network (FPN), which ensures the addition of features with minimal semantic gaps. It is critical to verify if the naive FPN structure could work in our proposed decoder structure. Therefore, it is necessary to conduct several analysis experiments to address these questions and provide a deeper insight into the proposed model.

We perform different experiments on the Mini COCO dataset and validate them on the original COCO validation set. Mini COCO is a subset of the COCO2017 training dataset which contains only 25K images. It is 10 times smaller than original MS COCO dataset[13]. Elaborate experiments are conducted in [19], confirming that detection models trained on Mini COCO had similar trends compared to those trained original dataset so that we can use it to efficiently conduct ablation experiments.

Integrity of multi-level structure Firstly, to verify the improvements truly come from multi-level design rather than simply from additional information from backbone feature maps, we design a structure with mixes the multi-scale at once, which denoted in Fig.5 (a). We use the same maxpooling operation to reshape features, then we add them simultaneously to generate one feature map and feed it to all encoder layers, to verify if it can boost our detection accuracy. The result is shown in Table.2, we do not observe any improvement, instead, the accuracy dropped to 31.2% compared to baseline DN-DETR (33.4%, trained with minicoco dataset). Thus simply stacking the information from the backbone is harmful for object queries to detect objects.



(a) DETR decoder with mixed features

(b) FPN style feature fusion

Figure 6: Comparision between our proposed multi-scale method and traditional multi-scale method in DETR

Effect of feature fusion module Then, to testify our feautre fusion design, we set a comparison experiment that strictly follows the pipeline of the FPN(as shown in Fig.5.(b), we fuse high-level feature $C5$ with features generated by the encoder, then fusing with downsampled $C4$ and $C3$ sequentially, to ensure we only adding features with similar semantic information. Although this structure contains multi-level features and seems to have minimum changes over our proposed one, it does not show a boost toward accuracy. In fact, compared to baseline DN-DETR, the detection accuracy is almost the same. Consequently, based on our analysis experiments, we can conclude that our feature fusion module is effective and feeding features with different semantic information into different decoder level is critical in our design. Moreover, it also distinct our work from works that involved FPN-like architecture like RT-DETR and CF-DETR.

Other Ablation study Feature map (C5) is fused at last decoder level to enhance semantic information of features. It takes most of our computational resources in proposed feature fusion models given the large number of channels of $C5$. Without it, the total number of parameters drops from 44.3M to 43.8M, which is almost the same as our baseline model (43.7M for DN-DETR). Moreover, the feature of C5 is semantically strong which might introduce redundant information because we already have the semantically strong base feature. So we tried to remove the feature fusion from the last level to see how it im-

Model	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	params
DN-DAB-DETR	33.4	53.2	34.7	13.9	35.8	51.1	43.7M
MLD-DETR	34.6	54.1	36.1	15.6	37.3	51.3	44.3M
Decoder w/ mixed features	32.2	51.5	33.9	13.3	35.1	49.9	44.3M
Decoder w/ FPN	32.7	52.6	34.2	47.7	35.6	15.2	44.3M
MLD-DETR(C5 eliminated)	34.2	53.8	36.0	15.3	37.2	50.3	43.8M

Table 2: Ablation Studies on Mini COCO.

pacts our accuracy. From the results of ablation experiments, feature fusion in the last two decoder layers is somehow critical: the detection accuracy will drop from 34.6% to 34.2%, counting almost 1/3 of our accuracy improvement. Moreover, we can observe that the detection accuracy for large objects dropped significantly, which implies the semantic information from C5 is critical for large object detection.

4.5 Limitations

Effects of more decoder layers Our proposed method solves the inefficiency in the transformer decoder as we introduces abundant information into it. So a natural question after solving this issue is that could we add more decoder layers to improve the detection accuracy. To answer this question, we did another ablation experiment which expanded the number of decoder layers from 6 to 9: we added one layer to each different decoder level (three F1 layers, three F2 layers, and three F3 layers). However, the detection transformer can not benefit from more decoder layers, the accuracy of the 9-layer decoder we have observed is almost the same as the one with 6 layers.

Extend model into multi-scale Our proposed model does not contain feature maps at different scale. Although reshaping feature maps into small ones are good for efficiency, there is no harm in introducing multi-scale features maps in different decoder level given the length of tokens in the DETR decoder is relatively short. Furthermore, models involved large multi-scale feature maps achieve much better accuracy than their single-scale counterpart. Thus we try to implement multi-scale features containing high-resolution features and feature pyramids. Firstly, we upsampled the base image features four times to their original spatial size, adding them with original C3. In this way, feature maps at bottom of the pyramid can be generated. Then we down-sample the generated map and add them to C4. So on and so forth we can get a whole feature pyramid which is then fed into different decoder level. However, scaled features do not provide better accuracy as they usually do. The overall performance of our scaled model is even worse compared to the baseline, which is far below than our proposed baseline model. There could be multiple reasons for the performance drop: 1. The size of feature maps in the last layer could be critical. Even though we generate a large feature map at the first level, it is downsampled to a smaller size later, so the final output cannot benefit from the large feature map. 2. Proposed feature fusion might not be optimal: The base image feature is upsampled to 4 times than its original and then fused with other features, so the semantic information could be damaged during these processes. However, due to the constrain of the hardware, we have not conducted any further experiments for scaled model, so we encourage the community to explore more possibilities on this related topic.

Model	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	params
MLD-DETR	34.6	54.1	36.1	15.6	37.3	51.3	44.3M
MLD-DETR w/ 9 layers	34.5	53.7	35.9	15.5	37.1	51.0	48.0M
Scaled-MLD-DETR	32.3	52.1	34.2	15.2	35.5	49.0	49.0M

Table 3: Experiments with more decoder layers and scaled feature maps. Note that experiments shown in this table also conducted on Mini COCO

5 Conclusion

In this paper, we introduce Detection Transformer with Multi-Level Decoder. It fuses multi-scale features and arrange them vertically at different layers of decoder, improving the efficiency at transformer decoder side. Furthermore, it provide a new perspective to the application of multi-scale features in DETR model. The proposed method improve the detection accuracy by 1.4% than its baseline DN-DAB-DETR on MS COCO dataset with almost no extra computational cost. Nevertheless, we found out our model does not work as expectation with scaled feature maps or more decoder layers. We anticipate that the DETR research community will explore more on related topics.

References

[1] Xipeng Cao, Peng Yuan, Bailan Feng, and Kun Niu. Cf-detr: Coarse-to-fine transformers for end-to-end object detection. In *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, 2022.

[2] 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 (ECCV)*, 2020.

[3] 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.

[4] Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

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

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] DH Hubel and TN Wiesel. Receptive fields of optic nerve fibres in the spider monkey. *J Physiol*, 154(3):572–580, 1960.

[9] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In *ECCV*, 2018.

[10] Feng Li, Ailing Zeng, Shilong Liu, Hao Zhang, Hongyang Li, Lei Zhang, and Lionel M Ni. Lite detr: An interleaved multi-scale encoder for efficient detr. *arXiv preprint arXiv:2303.07335*, 2023.

[11] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dndetr: Accelerate detr training by introducing query denoising. *arXiv preprint arXiv:2203.01305*, 2022.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[13] 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.

[14] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[15] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[16] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[17] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. *arXiv preprint arXiv:2108.06152*, 2021.

[18] S. Ren, Kaiming He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[19] N. Samet, S. Hicsonmez, and E. Akbas. Houghnet: Integrating near and long-range evidence for bottom-up object detection. In *ECCV*, 2020.

[20] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2567–2575, 2022.

[21] 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.

[22] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2022.

[23] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, 2019.

[24] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.