2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

A Method that Utilizes Negative Queries in Object Detection

Jue Zhou[1], Hong Chen[1] and Qingling Zhao[1]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing City 210094, China
🖂123106010660@njust.edu.cn

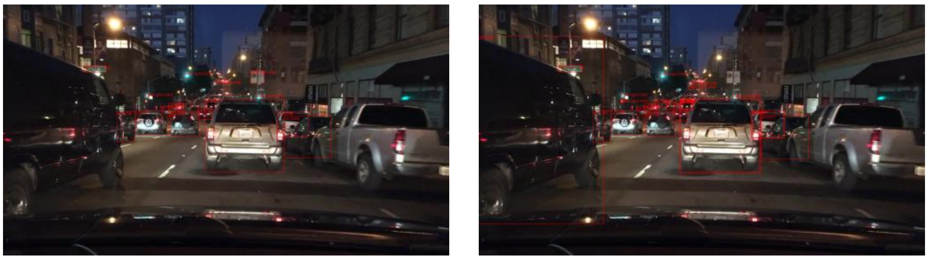
Abstract. DEtection TRansformer (DETR) represents an end-to-end paradigm for object detection based on transformers that generates multiple object queries per ground truth box and selects the best prediction through matching. However, studies have shown that many object queries rejected by the Hungarian matching algorithm still focus on foreground elements. Leveraging these potentially useful negative queries in DETR has thus emerged as a promising research direction.In this paper, I propose FUSION-DETR, a novel model that introduces a one-to-many matching strategy by grouping queries while preserving DETR's traditional one-to-one matching. This approach utilizes the foreground information embedded in queries rejected by Hungarian matching as negative samples. Furthermore, during training, the model dynamically assigns weights to the one-to-many loss using a clustering-based method, enhancing its robustness.Experiments demonstrate that the FUSION-DETR approach improves Deformable-DETR by 3.0 mAP50 on the BDD-100K datasets and achieves 70.5 mAP50 on COCO2017, outperforming existing DETR-based models incorporating one-to-many assignment.

Keywords: object recognition,neural networks,deep learning.

1 Introduction

1.1 Background

DEtection Transformer [3](DETR) represents an end-to-end paradigm for object detection that integrates a CNN-based feature extraction network, a Transformer encoder, and a Transformer decoder[2]. The decoder comprises self-attention mechanisms, cross-attention modules, and feedforward neural networks (FFNs), followed by classification and bounding box regression layers. In traditional DETR models [5], the decoder generates multiple object queries per real object and selects the best prediction through Hungarian matching. However, studies [20] have shown that many object queries rejected by the Hungarian matching algorithm still focus on foreground elements, containing rich and valuable information. This observation suggests that leveraging these rejected queries could further enhance model performance. Consequently, a one-to-many assignment strategy has been proposed to incorporate these queries into the optimization process.



(a)

(b)

Figure 1: (a) shows the detection performance of the baseline model prior to applying improvements, and (b) displays the detection results of the improved model. A comparison between the two figures reveals that the improved model demonstrates superior performance, especially in detecting small objects and overlapping objects. This observation indicates that the model has effectively leveraged the information carried by the negative queries to enhance detection accuracy in challenging scenarios.

The one-to-many assignment strategy has been extensively applied in CNN-based object recognition models [14], demonstrating its effectiveness in multi-class object detection. Within the DETR framework, two primary approaches have been explored for implementing one-to-many assignment:

(1) Grouping Object Queries:This approach[4] partitions the object queries into multiple groups and performs group-wise one-to-one assignment, enabling a real object to be assigned to multiple predictions. This strategy encourages high-confidence predictions for the real object while suppressing redundant predictions within the same group.

(2) Introducing Additional Query Branches: This approach [9] retains DETR’s one-to-one matching while incorporating additional query branches or decoder weights to facilitate one-to-many assignments. By expanding the query space, this approach enables each positive sample to be assigned multiple meaningful foreground queries

Both approaches offer distinct advantages and challenges: The first method is akin to ensemble learning [15], as it enhances object query representations through automatic learning without increasing the number of network parameters. It provides stronger supervision during training, leading to performance improvements in a simple and effective manner. However, it introduces several challenges:

(1) Reduction in Query Variable Dimensionality: Constraining the query space may limit model performance [16].

(2) Achieving One-to-Many Matching: Two key concerns arise here:

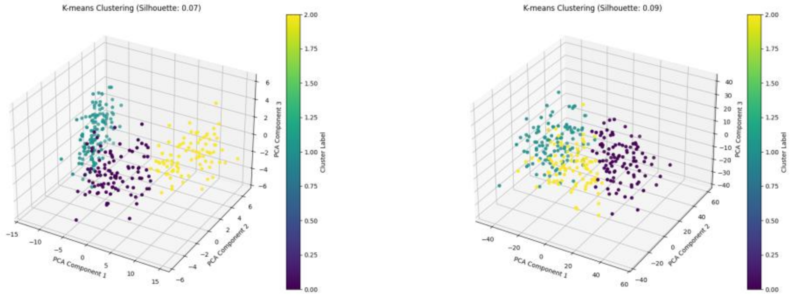
1): Without pre-trained weights: shared decoder weights across query groups may lead to similar spatial distributions, making it difficult to establish meaningful one-to-many relationships.

2): With pre-trained weights: grouping queries effectively is nontrivial. Ideally, queries within a group should focus on distinct targets while maintaining similar priority levels. However, coarse query grouping can undermine the benefits of pre-training, complicating convergence.

The second approach introduces additional one-to-many query branches while preserving the original DETR architecture. This method retains most of DETR's advantages, such as avoiding Non-Maximum Suppression (NMS) [17], which is beneficial for accuracy and efficiency. By assigning multiple foreground queries to a single real object globally, this method provides sufficient localization supervision. However, there are notable drawbacks:

(1) Loss Function Complexity: This method introduces two types of loss: the one-to-one loss and the one-to-many loss. These losses must be balanced, with the appropriate weights varying depending on the datasets. Currently, there is no systematic method to determine the correct weighting.

(2) Increased Training Time: The introduction of additional training weights leads to longer training cycles.



(a)

(b)

Figure 2: (a) shows the clustering diagram of the query vectors before passing through the decoupling module, while (b) displays the clustering diagram of the query vectors after passing through the decoupling module. Although the model's accuracy has reached a high level (43.3 mAP50), the silhouette score of the query vectors increases from 0.07 to 0.09 after decoupling. This increment demonstrates that the query decoupling module effectively helps the query vectors focus more on different targets, strengthening the model's ability to distinguish among individual objects with greater precision.

1.2 Our Methods

To address these challenges, I propose FUSION-DETR, which integrates the following three components:

Table 1: The results of the performance of improved deformable-detr on the COCO2017datasets.

model	queries	epoches	mAP	mAP50	APs	APm	APl
Conditional DETR[12]	300	108	43	64	22.7	46.7	61.5
Anchor DETR[8]	300	50	42.1	63.1	22.3	46.2	60
Efficient DETR[6]	300	50	45.1	63.1	28.3	48.4	59
DAB DETR[7]	900	50	45.7	66.2	26.1	49.4	63.1
Deformable DETR[22]	300	50	46.9	65.6	29.6	50.1	61.6
Group DETR[4]	900	12	50.1	-	32.4	53.2	64.7
Hybirds DETR[9]	900	12	50.6	68.7	34.4	53.9	63.5
FUSION DETR	**300**	**20**	**48**	**70.5**	**33.4**	**52**	**64.3**

(1) Joint Optimization of One-to-Many and One-to-One Losses: Building upon the one-to-one loss, I introduce the one-to-many loss, drawing inspiration from Group-DETR [4]. By jointly optimizing these two losses, I enable the model to simultaneously learn global detection accuracy while capturing local details, as shown in Figure.1.

(2) Query Decoupling Module:Inspired by the use of residual networks in deep learning, I introduce a query decoupling module. Query vectors pass through this module before the one-to-many loss computation, ensuring that queries within each group focus on different targets, thereby stabilizing the assignment between query vectors and ground truth boxes (as shown in Figure.2).

(3) Clustering-Based Weighting Method: I introduce a clustering-based weighting method that dynamically adjusts loss function weights on the basis of the clustering results of the target objects.Such an adjustment guides the model to allocate greater representational capacity to the primary task.

As evidenced by Table 1, by comparing the performance of FUSION-DETR with mainstream DETR models and two DETR models [4,9] that incorporate one-to-many assignment on the COCO2017 datasets, we can see that FUSION-DETR achieves 70.5 mAP50 with only 300 query dimensions, outperforming existing methods in many metrics.

2 Related Work

2.1 Transformer-based Object Detectors

Transformer-based object detectors, such as DETR, have revolutionized computer vision by adapting transformer’s originally designed for natural language processing (NLP) to object detection tasks. DETR replaces traditional components like region proposal networks (RPN) and non-maximum suppression (NMS)[17] with an end-to-end transformer architecture. By leveraging self-attention, DETR models the relationships between objects and background, which improves performance, especially in complex detection scenarios. DETR, however, is subject to several challenges, including slow training times and difficulty in detecting small objects, which has prompted the development of more efficient variants.

Several notable improvements have been made to DETR. Deformable-DETR [22] introduces a sparse attention mechanism and learnable sampling points to lower the computational burden while improving the model's capability to focus on relevant regions. Lite-DETR [10] designs an efficient encoder module that alternately updates high-level features, streamlining the model. MS-DETR [20] incorporates a combination of one-to-one and one-to-many supervision to guide the main decoder’s queries, enhancing detection accuracy. CO-DETR [23] improves the efficiency and power of DETR-based detectors by diversifying the label assignment strategy, addressing some of the model’s limitations.

2.2 Stable Assignment Between Queries and GT Boxes

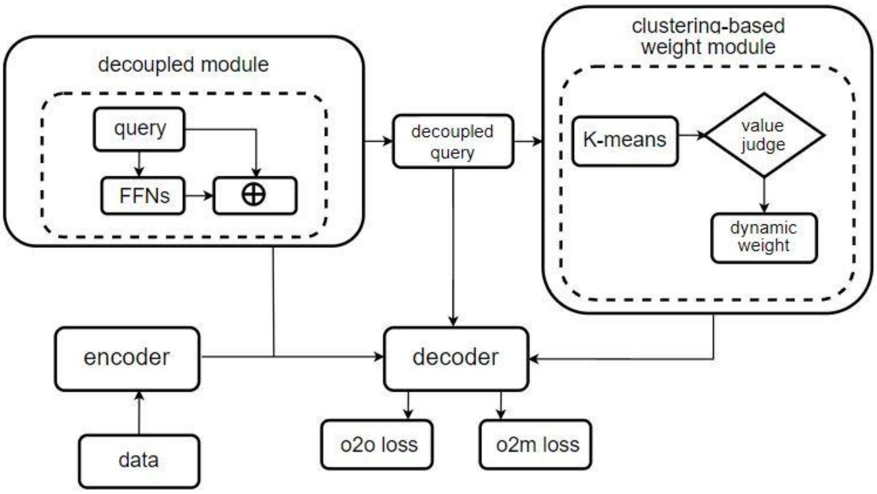


Figure 3. Algorithm flowchart.

DETR leverages the Hungarian algorithm for the one-to-one association of queries with ground truth (GT) boxes, ensuring a one-to-one correspondence between each ground truth (GT) box and a single query. Queries that do not match any GT box are classified as background. However, in complex scenarios such as small objects or overlapping targets, some queries may struggle to find a stable match with the GT boxes, resulting in unstable assignments.

Several methods have been proposed to stabilize the matching process. Deformable-DETR [22] addresses this by grouping queries and allowing multiple queries to match the same GT box from different perspectives, improving stability in challenging detection cases. DN-DETR [11] generates pseudo-labels through multiple perturbations of the object and matches them with GT boxes to enhance the robustness and stability of the matching process. SAM-DETR [19] presents a plug-in module that projects both object queries and encoded image features into a unified feature embedding space, thereby enabling more consistent and stable query-to-box associations.

3 Mechanism of FUSION-DETR

3.1 Hybrid Loss Training

During the training phase, the FUSION-DETR model is trained by combining both one-to-one and one-to-many assignment losses. These two losses share the same decoder module for their computation. The one-to-one loss is generated using the traditional DETR approach, where every ground truth (GT) box corresponds to a single query.

As the Figure.3 shows,for the generation of the one-to-many assignment loss, the query vectors are first processed by a query decoupling module. This module ensures that queries within different groups focus more on their respective independent objects, improving the model's capacity to differentiate between multiple foreground objects. After decoupling, the adjusted query vectors are passed into the decoder, where they are processed to produce class predictions and bounding box predictions through two separate prediction heads.

3.2 Clustering-Based Weighting Method

Through extensive experimentation, it was observed that once the model's performance reaches a certain threshold, increasing the weight of the one-to-many loss can significantly accelerate convergence. To achieve this,as the Figure.3 shows, FUSION-DETR utilize the k-means algorithm to cluster the adjusted query vectors. The silhouette coefficient, which quantifies clustering quality, is subsequently computed. If the silhouette coefficient is below a predefined threshold (denoted as δ), the weight of the one-to-many loss is increased to λ_2. Otherwise, it is set to λ_1, promoting more dynamic and context-sensitive adjustments to the loss function during training.

3.3 Mathematical Expression

Original DETR Loss. In the original DETR model, the overall loss function is composed of two key elements: classification loss and bounding box loss. The final loss is a weighted sum of these two components, calculated over all queries.

Let $Q = \{q_1, q_2, \dots, q_N\}$, be the set of queries, and $G = \{g_1, g_2, \dots, g_N\}$ be the set of ground truth objects, where N is the number of queries and ground truth objects.

The classification loss L_{cls} is computed using cross-entropy between predicted class labels $\hat{p}_i$ and ground truth labels y_i :

$$L_{\text{cls}} = -\sum_{i=1}^{N} 1_{\{y_i \neq \text{no\_obj}\}} \log(\hat{p}_i)$$

The bounding box loss L_{bbox} is computed using the L_1 loss between predicted bounding boxes $\hat{b}_i$ and the ground truth boxes b_i , plus a generalized intersection-over-union (GIoU) loss:

$$L_{\text{bbox}} = \sum_{i=1}^{N} 1_{\{y_i \neq \text{no\_obj}\}} \left(\text{L1}(\hat{b}_i, b_i) + \text{GIoU}(\hat{b}_i, b_i) \right)$$

The overall DETR loss is defined as the weighted sum of the classification loss and the bounding box regression loss:

$$L_{\text{DETR}} = L_{\text{cls}} + \lambda_{\text{bbox}} L_{\text{bbox}}$$

where λ_{bbox} is a hyper parameter to balance the two losses.

Optimized DETR Loss with One-to-Many Loss:The optimized DETR model introduces the one-to-many loss, which involves splitting the queries into multiple groups, each of which is matched independently to ground truth objects using the Hungarian algorithm. The optimization procedure involves two loss components:one-to-one loss and one-to-many loss.

The query decoupling module processes the initial set of query vectors and outputs an adjusted set of query vectors $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_N\}$, where each query vector is decoupled to focus more distinctly on a specific group of objects. The operation of the query decoupling module is denoted by the function f_{decouple}:

$$\tilde{Q} = f_{\text{decouple}}(Q) = Q + \text{FFN}(\text{LN}(Q))$$

where f_{decouple} is the function that adjusts the query vectors to ensure they focus on different sets of objects. The nature of this function can be tailored to the architecture and task at hand (e.g., a learnable transformation, attention mechanism, or clustering-based approach).

Let $\tilde{Q} = \{\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_N\}$ be the adjusted set of query vectors, which are grouped into M clusters. The m-th group of queries is $\tilde{Q}_m = \{\tilde{q}_m^1, \tilde{q}_m^2, \ldots, \tilde{q}_m^{N_m}\}$.

The one-to-one loss $L_{\text{1-to-1}}(\tilde{q}_m^i, y_m^i, b_m^i)$ is computed for each query within each group using the Hungarian matching. The one-to-one loss formula is:

$$L_{\text{1-to-1}}(\tilde{q}_m^i, y_m^i, b_m^i) = 1_{\{y_m^i \neq \text{no\_obj}\}} \left(\text{CE}(\hat{p}_m^i, y_m^i) + \text{L1}(\hat{b}_m^i, b_m^i) \right)$$

The total one-to-one loss for group m is:

$$L_{\text{1-to-1}}(\tilde{Q}_m) = \sum_{i=1}^{N_m} L_{\text{1-to-1}}(\tilde{q}_m^i, y_m^i, b_m^i)$$

The one-to-many loss $L_{\text{1-to-many}}$ is computed by weighting the one-to-one loss of each group with a dynamic coefficient:

$$L_{\text{1-to-many}} = \sum_{m=1}^{M} \alpha_{\text{1-to-many},m} \cdot L_{\text{1-to-1}}(\tilde{Q}_m)$$

We figure out the structure of $\tilde{Q}$ that $\tilde{Q} \in \mathbb{R}^{B \times (N_m \times M) \times D}$,where B is the batch size, N_m is the number of queries per group,M is the number of groups, and D is the feature dimension. We first treat it as a concatenation of M subgroups and $\tilde{Q_m} \in \mathbb{R}^{N_m \times (B \times D)}$ .The silhouette coefficient SC_m is used to adjust the weight $\alpha_{\text{1-to-many},m}$ of the one-to-many loss for each group m. The silhouette coefficient is calculated as:

$$\text{SC}_m(\tilde{Q}_m) = \frac{1}{M} \sum_{i=1}^{M} \frac{b_i - a_i}{\max(a_i, b_i)}$$

where a_i represents the mean distance to other samples within the same cluster, and b_i represents the mean distance to the nearest cluster.

Based on the silhouette coefficient, the weight adjustment rule is:

$$\alpha_{\text{1-to-many},m} = \begin{cases} \lambda_1, & \text{if } \mathrm{SC}_m(\tilde{Q}_m) < \delta \\ \lambda_2, & \text{if } \mathrm{SC}_m(\tilde{Q}_m) \geq \delta \end{cases}$$

Finally, the total loss for the optimized DETR model is the sum of the original DETR loss and the weighted one-to-many loss:

$$L_{\text{final}} = L_{\text{DETR}} + L_{\text{1-to-many}}$$

4 Experiments

4.1 Experimental Setup and Details

Datasets:The experiments were conducted using the challenging BDD100K[18] datasets and COCO2017 datasets[13]. They are all formatted in the COCO format. Model performance was evaluated using standard metrics, including mAP(mean Average Precision) and others.

Table 2: The results of the models' performance before and after the improvements of FUSION-DETR with the training parameters fixed.

model	datasets	epoches	mAP	mAP50	APs	APm	APl
focus-detr[21]	bdd-10K	50	20.4	39.3	10.3	28.2	39.4
focus-detr with FD	**bdd-10K**	**20**	**22.3**	**43.5**	**11.7**	**31.8**	**43.4**
deformable-detr[22]	bdd-10K	50	22	43.4	12.7	31.8	43.4
deformable-detr with FD	**bdd-10K**	**20**	**22.7**	**44.7**	**12.5**	**32.4**	**43.9**
deformable-detr	bdd-100K	80	20.5	41.3	11.8	31	41.3
deformable-detr with FD	**bdd-100K**	**30**	**23.8**	**44.3**	**12.4**	**32.3**	**43.4**
deformable-detr	COCO2017	50	46.9	65.6	29.6	50.1	61.6
deformable-detr with FD	**COCO2017**	**30**	**48**	**70.5**	**33.4**	**52**	**64.3**

DETR models:The FUSION-DETR method was applied to improve two existing DETR models: Deformable-DETR and Focus-DETR. These models were selected for their strong performance on large datasets with relatively low training costs, and because both lacked effective strategies for utilizing negative queries which can be compensated by the FUSION-DETR method.

Experimental Details: The performance of the two models before and after applying the FUSION-DETR method was compared on the BDD10K datasets. The training was performed on two NVIDIA 3080 GPU for 30 epochs. Additionally, the Deformable-DETR model was also evaluated on the BDD100K datasets and COCO2017 datasets, using a single NVIDIA 3090 GPU, with 30 epochs of training for improvement versions. The ResNet-50 architecture was used for feature extraction in all experiments, and the Adam optimizer was employed for training.

4.2 Main Results

Table 3: The results of the performance of improved deformable-detr on the bdd100K datasets in the ablation study.

model	epoches	mAP	mAP50	APs	APm	APl
Deformable-detr[22]	80	20.5	41.3	11.8	31	41.3
FD without decoupling module	20	22.5	42.4	11.5	30.8	41.5
FD without dynamic weight	20	22.7	42.8	11.5	31	42.2
FUSION-DETR	30	23.8	44.3	12.4	32.3	43.4

Table 4: The results of the performance of improved deformable-detr on the bdd100K datasets with different λ_2.

model	λ_2	epoches	mAP	mAP50	APs	APm	APl
improved d-d	0.1	80	23.2	43.3	12	31.5	42.6
improved d-d	0.2	80	23.8	44.3	12.4	32.3	43.4
improved d-d	0.3	80	23.1	43.2	11.9	31.5	42.5

This section presents the results from three different sets of experiments: (1)A comparison of model performance before and after improvements, while holding training parameters constant; (2) The relationship between batch size and the performance of the optimized model; (3) The effect of the one-to-many loss weight on model performance.

Comparison of Model Performance with Fixed Training Parameters:The results of this experiment are shown in Table.2. On the BDD10K datasets, the FUSION-DETR method improved the mAP50 score of Deformable-DETR by 1.3% and Focus-DETR (Zheng et al., 2023) by 4.2%. On the BDD100K datasets, FUSION-DETR improved the mAP50 score of Deformable-DETR by 3.0%. On the COCO2017 datasets, FUSION-DETR improved the mAP50 score of Deformable-DETR by 4.9%. These results demonstrate that the FUSION-DETR method can effectively enhance the performance of DETR-based models and exhibits a certain degree of robustness across datasets.

Ablation Study:The experimental results are shown in Table 3.To evaluate the impact of the dynamic weighting mechanism and the decoupling module on the model’s performance, I conducted comparative experiments on the BDD-100K datasets.

In this study:Removing the decoupling module means that the query vectors are fed directly into the decoder without undergoing decoupling. Not using the dynamic weighting method means that the weight of the one-to-many loss remains fixed throughout the entire training process.

The results show that both the decoupling module and the clustering-based dynamic weighting mechanism contribute to improving model performance to some extent. Specifically, the decoupling module increases the model's mAP50 by 1.1, while the dynamic weighting mechanism further boosts it by 1.5. When these two components are combined, they achieve even better performance and increase the model's mAP50 by 3.0, demonstrating their complementary benefits in enhancing detection accuracy

Effect of One-to-Many Loss Weight:In this experiment, the value of λ_1 was kept constant during the first stage, and different values for λ_2 were selected to train the improved Deformable-DETR on the BDD100K datasets. The results are presented in Table.4. It was observed that selecting an appropriate value for λ_2 is critical, as both excessively high and low values lead to a decline in model performance. This demonstrates the importance of fine-tuning the weight of the one-to-many loss to achieve optimal results.

5 Conclusion

In this study, I propose a method for fully leveraging the negative queries in DETR. This approach was implemented to enhance the performance of two DETR-based models, DeformableDETR and Focus-DETR, and was evaluated on the COCO2017 and BDD100K datasets. Experimental results confirm the efficacy of the proposed method. It is hoped that the FUSION-DETR method will contribute to further advancements in the field and inspire future research in object detection using transformer-based architectures.

References

1. Ahmed, M., Seraj, R., and Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8):1295.
2. Ashish, V. (2017). Attention is all you need. Advances in neural information processing systems, 30:I.
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020).End-to-end object detection with transformers. In European conference on computer vision, pages 213–229. Springer.
4. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng,G., and Wang, J. (2023). Group detr: Fast detr training with group-wise one-to-many assignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6633–6642.
5. Dai, Z., Cai, B., Lin, Y., and Chen, J. (2021). Up-detr: Unsupervised pre-training for object detection with transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1601–1610.

6. Yao Z, Ai J, Li B, et al. Efficient detr: improving end-to-end object detector with dense prior[J]. arXiv preprint arXiv:2104.01318, 2021.
7. Liu S, Li F, Zhang H, et al. Dab-detr: Dynamic anchor boxes are better queries for detr[J]. arXiv preprint arXiv:2201.12329, 2022.
8. Wang Y, Zhang X, Yang T, et al. Anchor detr: Query design for transformer-based detector[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 2567-2575.
9. Jia, D., Yuan, Y., He, H., Wu, X., Yu, H., Lin, W., Sun, L., Zhang, C., and Hu, H. (2023). Detrs with hybrid matching. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 19702–19712.
10. Li, F., Zeng, A., Liu, S., Zhang, H., Li, H., Zhang, L., and Ni, L. M. (2023). Lite detr: An interleaved multi-scale encoder for efficient detr. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 18558–18567.
11. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L. M., and Zhang, L. (2022). Dn-detr: Accelerate detr training by introducing query denoising. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 13619–13627.
12. Meng D, Chen X, Fan Z, et al. Conditional detr for fast training convergence[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 3651-3660.
13. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Computer vision-ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13, pages 740–755. Springer.
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer.
15. Mohammed, A. and Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University-Computer and Information Sciences, 35(2):757–774.
16. Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR.
17. Wu, G. and Li, Y. (2021). Non-maximum suppression for object detection based on the chaotic whale optimization algorithm. Journal of Visual Communication and Image Representation, 74:102985.
18. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2636–2645.
19. Zhang, G., Luo, Z., Yu, Y., Cui, K., and Lu, S. (2022). Accelerating detr convergence via semantic-aligned matching. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 949–958.
20. Zhao, C., Sun, Y., Wang, W., Chen, Q., Ding, E., Yang, Y., and Wang, J. (2024). Msdetr: Efficient detr training with mixed supervision. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 17027–17036.
21. Zheng, D., Dong, W., Hu, H., Chen, X., and Wang, Y. (2023). Less is more: Focus attention for efficient detr. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6674–6683.

22. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2010). Deformable detr: Deformable transformers for end-to-end object detection. arxiv 2020. arXiv preprint arXiv:2010.04159, 3.
23. Zong, Z., Song, G., and Liu, Y. (2023). Detrs with collaborative hybrid assignments training. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6748–6758.