RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer

Technical Report

Wenyu Lv[1] **Yian Zhao**[2] **Qinyao Chang**[1] **Kui Huang**[1] **Guanzhong Wang**[1] **Yi Liu**[1]

[1]Baidu Inc. [2]Peking University Shenzhen Graduate School

lvwenyu01@baidu.com zhaoyian@stu.pku.edu.cn

Abstract

In this report, we present RT-DETRv2, an improved Real-Time DEtection TRansformer (RT-DETR). RT-DETRv2 builds upon the previous state-of-the-art real-time detector, RT-DETR, and opens up a set of bag-of-freebies for flexibility and practicality, as well as optimizing the training strategy to achieve enhanced performance. To improve the flexibility, we suggest setting a distinct number of sampling points for features at different scales in the deformable attention to achieve selective multi-scale feature extraction by the decoder. To enhance practicality, we propose an optional discrete sampling operator to replace the `grid_sample` operator that is specific to RT-DETR compared to YOLOs. This removes the deployment constraints typically associated with DETRs. For the training strategy, we propose dynamic data augmentation and scale-adaptive hyperparameters customization to improve performance without loss of speed. Source code and pre-trained models will be available at https://github.com/lyuwenyu/RT-DETR.

1 Introduction

Object detection is a fundamental vision task that involves identifying and localizing objects in an image. Among them, real-time object detection is an important field and has a wide range of applications, such as autonomous driving (Atakishiyev et al. [2024]). With the development of the last few years, YOLO detectors (Redmon and Farhadi [2017, 2018], Bochkovskiy et al. [2020], Glenn. [2022], Xu et al. [2022], Li et al. [2023], Wang et al. [2023], Glenn. [2023], Wang et al. [2024a,b]) are without doubt the most prestigious framework in this field. The reason for this is the reasonable balance achieved by the YOLO detectors.

The advent of RT-DETR (Zhao et al. [2024]) opens up a new technological avenue for real-time object detection, breaking the dependency on the YOLO in this field. RT-DETR proposes an efficient hybrid encoder to replace the vanilla Transformer encoder in DETR (Carion et al. [2020]), which significantly improves the inference speed by decoupling the intra-scale interaction and cross-scale fusion of multi-scale features. To further improve the performance, RT-DETR proposes the uncertainty-minimal query selection, which provides high-quality initial queries to the decoder by explicitly optimizing the uncertainty. Moreover, RT-DETR provides a wide range of detector sizes and supports flexible speed tuning to accommodate various real-time scenarios without retraining. RT-DETR represents a novel, end-to-end, real-time detector that marks a significant advancement for the DETR family.

In this report, we present RT-DETRv2, an improved real-time detection Transformer. This work is built upon the recent RT-DETR and opens up a set of bag-of-freebies for flexibility and practicality within the DETR family, as well as optimizing the training strategy to achieve enhanced performance. Specifically, RT-DETRv2 suggests setting a distinct number of sampling points for features at different scales within the deformable attention module to achieve selective multi-scale feature extraction by the decoder. In the realm of enhancing practicality, RT-DETRv2 provides an optional discrete sampling operator to replace the original `grid_sample` operator, which is specific to DETRs, thus eliminating the deployment constraints typically associated with detection Transformers. Furthermore, RT-DETRv2 optimizes the training strategy, including dynamic data augmentation and scale-adaptive hyperparameters customization, with the objective of improving performance without loss of speed. The results demonstrate that RT-DETRv2 provides an improved baseline with bag-of-freebies for RT-DETR, increases the flexibility and practicality, and the proposed training strategies optimize the performance and training cost.

2 Method

The framework of RT-DETRv2 remains the same as RT-DETR, with only modifications to the deformable attention module of the decoder.

2.1 Framework

Distinct number of sampling points for different scales. Current DETRs utilize the deformable attention module (Zhu et al. [2020]) to alleviate the high computational overhead caused by the long sequence of inputs composed of multi-scale features. The RT-DETR decoder retains this module, which defines the same number of sampling points at each scale. We argue that this constraint ignores the intrinsic differences in features at different scales and limits the feature extraction capability of the deformable attention module. Therefore, we propose to set distinct numbers of sampling points for different scales to achieve more flexible and efficient feature extraction.

Discrete sampling. To improve the practicality of the RT-DETR and to make it available everywhere. We focus on comparing the deployment requirements of YOLOs and RT-DETR, where the RT-DETR-specific `grid_sample` operator limits its broad applicability. Therefore, we propose an optional `discrete_sample` operator to replace the `grid_sample`, thus removing the deployment constraints of RT-DETR. Specifically, we perform a rounding operation on the predicted sampling offsets, omitting the time-consuming bilinear interpolation. However, the rounding operation is non-differentiable, so we turn off the gradient of the parameters used to predict the sampling offsets. In practice, we first employ the `grid_sample` operator for training and then replace it with the `discrete_sample` operator for fine-tuning. For inference and deployment, the model employs the `discrete_sample` operator.

2.2 Training Scheme

Dynamic data augmentation. To equip the model with robust detection performance, we propose the dynamic data augmentation strategy. Considering the poor generalizability of the detector in the early training period, we apply stronger data augmentation, while in the later training period we decrease its level to adapt the detector to the detection of the target domain. Specifically, we maintain the RT-DETR data augmentation in the early period, while turning off `RandomPhotometricDistort`, `RandomZoomOut`, `RandomIoUCrop`, and `MultiScaleInput` in the last two epochs.

Scale-adaptive hyperparameters customization. We also observe that the scaled RT-DETRs of different sizes are trained with the same optimizer hyperparameters, resulting in their sub-optimal performance. Therefore, we propose scale-adaptive hyperparameters customization for scaled RT-DETRs. Considering that the pre-trained backbone for light detector (*e.g.*, ResNet18 (He et al. [2016])) has lower feature quality, we increase its learning rate. On the contrary, the pre-trained backbone with large detector (*e.g.*, ResNet101 (He et al. [2016])) has higher feature quality and we decrease its learning rate.

3 Experiment

3.1 Implementation Details

As with RT-DETR, we use ResNet (He et al. [2016]) pretrained on ImageNet as the backbone and train RT-DETRv2 with the AdamW (Loshchilov and Hutter [2018]) optimizer with a batch size of 16 and apply the exponential moving average (EMA) with $ema\_decay = 0.9999$. For the optional discrete sampling, we first pre-train $6\times$ with the `grid_sample` operator and then fine-tune $1\times$ with the `discrete_sample` operator. For scale-adaptive hyperparameters customization, the hyperparameters are shown in Tab. 1, where *lr* represents the learning rate.

Table 1: The hyperparameters of RT-DETRv2.

Model	Backbone	$lr_{backbone}$	lr_{det}
RT-DETRv2-S	ResNet18	1e-4	1e-4
RT-DETRv2-M	ResNet34	5e-5	1e-4
RT-DETRv2-L	ResNet50	1e-5	1e-4
RT-DETRv2-X	ResNet101	1e-6	1e-4

3.2 Evaluation

RT-DETRv2 is trained on COCO (Lin et al. [2014]) `train2017` and validated on COCO `val2017` dataset. We report the standard AP metrics (averaged over uniformly sampled IoU thresholds ranging from $0.50 - 0.95$ with a step size of 0.05), and AP_{50}^{val} commonly used in real scenarios.

3.3 Results

The comparison with RT-DETR(Zhao et al. [2024]) is shown in Tab. 2. RT-DETRv2 outperforms RT-DETR at different scales of detectors without loss of speed.

Table 2: **Comparison of RT-DETR and RT-DETRv2.** The FPS is reported on T4 GPU with TensorRT FP16. For evaluation, all input sizes are fixed on 640 × 640.

Model	Backbone	Dataset	#Params (M)	$\text{FPS}_{bs=1}$	AP^{val}	AP^{val}_{50}
RT-DETR-S	ResNet18	COCO	20	217	46.5	63.8
RT-DETR-M	ResNet34	COCO	31	161	48.9	66.8
RT-DETR-M*	ResNet50	COCO	36	145	51.3	69.6
RT-DETR-L	ResNet50	COCO	42	108	53.1	71.3
RT-DETR-X	ResNet101	COCO	76	74	54.3	72.7
RT-DETRv2-S	ResNet18	COCO	20	217	47.9 (↑ **1.4**)	64.9 (↑ **1.1**)
RT-DETRv2-M	ResNet34	COCO	31	161	49.9 (↑ **1.0**)	67.5 (↑ **0.7**)
RT-DETRv2-M*	ResNet50	COCO	36	145	51.9 (↑ **0.6**)	69.9 (↑ **0.3**)
RT-DETRv2-L	ResNet50	COCO	42	108	53.4 (↑ **0.3**)	71.6 (↑ **0.3**)
RT-DETRv2-X	ResNet101	COCO	76	74	54.3 (↑ **0.0**)	72.8 (↑ **0.1**)

3.4 Ablations

Ablation on sampling points. We perform an ablation study on the total number of sampling points of the `grid_sample` operator. The total number of sampling points is calculated as `num_head` × `num_point` × `num_query` × `num_decoder`, where `num_point` represents the sum of sampling points for each scale feature in each grid. The results show that reducing the number of sampling points does not cause a significant degradation in the performance, *cf.* Tab. 3. This means that practical application is unlikely to be affected in most industrial scenarios.

Table 3: Ablation on sampling points.

Model	Sampling method	#Points	AP^{val}	AP^{val}_{50}
RT-DETRv2-S	`grid_sample`	86,400	47.9	64.9
RT-DETRv2-S	`grid_sample`	64,800	47.8	64.8 (↓ **0.1**)
RT-DETRv2-S	`grid_sample`	43,200	47.7	64.7 (↓ **0.2**)
RT-DETRv2-S	`grid_sample`	21,600	47.3	64.3 (↓ **0.6**)

Ablation on discrete sampling. We then remove the `grid_sample` and replace it with `discrete_sample` for the ablation. The results show that this operation does not cause a noticeable reduction in AP^{val}_{50}, but does eliminate the deployment constraints of the DETRs, *cf.* Tab. 4.

Table 4: Ablation on discrete sampling.

Model	Backbone	Sampling method	AP^{val}	AP^{val}_{50}
RT-DETRv2-S	ResNet18	`discrete_sample`	47.4	64.8 (↓ **0.1**)
RT-DETRv2-M	ResNet34	`discrete_sample`	49.2	67.1 (↓ **0.4**)
RT-DETRv2-M*	ResNet50	`discrete_sample`	51.4	69.7 (↓ **0.2**)
RT-DETRv2-L	ResNet50	`discrete_sample`	52.9	71.3 (↓ **0.3**)

4 Conclusion

In this report, we propose RT-DETRv2, an improved real-time detection Transformer. RT-DETRv2 opens up a set of bag-of-freebies to increase the flexibility and practicality of RT-DETR, optimizing the training strategy to achieve enhanced performance without loss of speed. We hope that this report will provide insights for the DETR family and broaden the scope of RT-DETR applications.

References

Shahin Atakishiyev, Mohammad Salameh, Hengshuai Yao, and Randy Goebel. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. *IEEE Access*, 2024.

Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Jocher Glenn. Yolov5 release v7.0. *https://github.com/ultralytics/yolov5/tree/v7.0*, 2022.

Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. Pp-yoloe: An evolved version of yolo. *arXiv preprint arXiv:2203.16250*, 2022.

Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3.0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*, 2023.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

Jocher Glenn. Yolov8. *https://github.com/ultralytics/ultralytics/tree/main*, 2023.

Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024a.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024b.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974, 2024.

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*, pages 213–229. Springer, 2020.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

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.