Enhanced Training of Query-Based Object Detection via Selective Query Recollection
- Supplementary Materials -

A. Implementation Detail

A.1. Query with Priors

Recent query-based object detectors associate priors with queries. These priors have multiple forms [1, 3, 4, 6], but generally, they are designed to involve spatial and scale priors which helps models converge faster. When implementing these methods, a query is usually regarded as two parts: one is the embedding that focuses on interacting with feature map and producing high-level object information, called *content*, another is similar to the concept *anchor* [5] which serves as a *reference* point for locating/scaling the object, and narrows down the range of feature-interaction (e.g., cross-attention).

For SQR, we recollect the *query* at each stage. That means both the content and the corresponding reference are recollected in the our operation.

B. Additional Experiments And Analysis

B.1. SQR with DN-DETR

Model	Epoch	AP	AP50	AP75
DN-DETR [2]	12e	38.5	58.8	40.6
SQR-DN-DETR	12e	**40.4**	61.1	42.7
DN-DETR [2]	50e	44.1	64.4	46.7
SQR-DN-DETR	50e	**45.2**	65.7	48.3

Table S1. SQR with DN-DETR

DN-DETR [2] is a training strategy that is based on DAB-DETR. We show that SQR is compatible with DN. Table S1 presents the results of DN-DETR w/ and w/o SQR. SQR enhances DN-DETR by $+1.9$ with 12 epochs schedule, and $+1.1$ with 50 epochs. We see that the benefit of SQR is less with extra long training epochs than with standard 1x schedule, although the improvement is still significant. Similar observations are obtained from Table 8 as Deformable DETR get $+2.7$ AP by SQR with 12 epochs while get $+1.4$ AP with 50 epochs. We elaborate this point in the following section.

B.2. SQR with 12e And 50e

Observation on Table S1 and Table 8 indicate that the benefit of SQR becomes less with extra long training epochs than with standard 1x schedule. However, the benefit cannot be replaced by extended training epochs, as already analyzed in Fig.5. DN-DETR is under-fitted at the 12th epoch because of its limited convergence speed and the multi-scale training setting, in this case, the mechanism of query recollection produces more supervision and speeds up the convergence of later stages. Under 50 epochs, as later stages get more supervisions and the model converges, the benefit of accelerated convergence becomes less, on the other hand, the benefit from the training emphasis and the mitigated cascading errors is less affected by the long schedule, which still brings strong improvement.

C. DQRR: DQR with Recurrence for Reducing Model Size

Herein, we explore an interesting direction enabled by Dense Query Recollection, i.e., using DQR to reduce model size. Existing methods typically have more than 6 decoding stages in decoder. Can we directly train a detector where all decoding stages share parameters? We implement this concept on vanilla Adamixer and find that the model is not able to converge. But we find DQR has the capability to achieve the goal.

As we know, a strong decoding stage at the end, i.e. the final stage, is obtained after training with DQR. This stage has seen every possibly intermediate queries that ever exist along the decoding process. A natural attempt is to replace all stages' parameters with the final stage's parameter during inference, forming a pathway as $\mathcal{PT}^{6\text{-}6\text{-}6\text{-}6\text{-}6\text{-}6}$. However, this results in a 0 AP result! The reason is that the output of stage 6 shifts from its input, so stage 6 cannot recognize its own output, thus, it applies random refinement (negative effect) on it.

To address the problem, during training, **we recollect the**

output of stage 6**, and feed back to itself as its input**. In such way, stage 6 gets chance to learn refining its output. Then, we recurrently use stage 6 only for inference. We name this method as DQRR (Dense Query Recollection and Recurrence). The result is shown in Table S2

Table S2. DQRR: Dense query recollection and recurrence.

# stage	AP	AP50	AP75
1	0.125	0.290	0.092
2	0.329	0.514	0.346
3	0.400	0.583	0.427
4	0.422	0.606	0.453
5	0.428	0.612	0.459
6	0.428	0.613	0.459

With DQRR, all decoding stages share the same parameters, so the model size is reduced by 70% (1.6GB to 513 MB). And it only needs 5 stages to achieves better performance than previous (42.8AP vs 42.5 AP).

D. Notation

The notation used in this paper is summarized in Table S3

References

[1] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5354–5363, 2022. 1

[2] Feng Li, Hao Zhang, Shi guang Liu, Jian Guo, Lionel M. Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13609–13617, 2022. 1

[3] Shilong Liu, Feng Li, Hao Zhang, Xiao Bin Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *ArXiv*, abs/2201.12329, 2022. 1

[4] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3631–3640, 2021. 1

[5] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 1

[6] Yingming Wang, X. Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI Conference on Artificial Intelligence*, 2022. 1

Table S3. Notation in this paper by the order of appearance.

Notation	Definition
QR	Query Recollection
SQR	Selective Query Recollection
q_i^0	The i_{th} initial query
n	Total number of initial query
N	The set {1,2,3, ..., n}
q_i^1	The output of the first stage refining q_i^0, also known as the i_{th} query at stage 1
s	The index of stage
D^s	The s_{th} decoding stage
q_i^s	The i_{th} query at stage s
q^s	The set of queries $q^s = \{q_i^s \mid i \in N\}$
x	Image features
$(\mathcal{A} \circ \mathcal{F})$	Self- and cross-attention and feed forward network
P_i^s	Prediction of q_i^s
G	A ground-truth
IoU	Intersection over Union
TP	True-positive
FP	False-positive
DQR	Dense Query Recollection
q	A set of queries $\{q_i \mid i \in \{1, 2, ..., n\}\}$, as a basic unit
$\hat{q}$	q that requires supervision during training
$\mathcal{PT}$	Pathway
C	A collection of q
$\#Supv$	Number of supervision
$\mathbb{R}$	Removal Probability
DQRR	Dense query recollection and recurrence