PR-DETR for Efficient and Lightweight Object Detection via Weight-sharing Softmask Pruning and Spatial Reduction

1st Minjie Zhang
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, China
zhangminjie200112@163.com

2nd Hairui Ye
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, China
yehairui99@163.com

3rd Yunfei Tong
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, China
10182415@mail.ecust.edu.cn

4th Zhe Wang*
Department of Computer Science and Engineering
East China University of Science and Technology
Shanghai, China
wangzhe@ecust.edu.cn

***Abstract*—DETR achieves competitive detection accuracy but suffers from high computational cost and large parameter size, limiting deployment on mobile and edge devices. In this paper, we propose PR-DETR, a lightweight detection transformer that combines weight-sharing softmask pruning with spatial reduction token compression to improve efficiency while preserving accuracy. The proposed weight-sharing softmask pruning strategy enables differentiable structural pruning across MLP, attention, and convolution blocks with flexible control of the pruning budget. In addition, the spatial reduction scheme compresses key and value tokens in attention to further reduce computation. Experiments on VOC0712 and MiniCOCO2017 demonstrate that PR-DETR achieves substantial reductions in parameters and FLOPs while maintaining comparable mAP to existing lightweight DETR models.**

***Index Terms*—object detection, detection transformer, model compression, structured pruning, spatial reduction**

I. Introduction

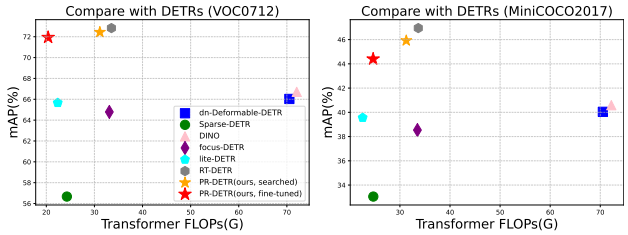
Object detection stands as a pivotal task within the realm of computer vision. In recent times, deep learning-driven object detection has witnessed remarkable advancements. Notably, the DETR [1] model, leveraging the power of Transformers, has achieved outstanding results in object detection due to its end-to-end design and robust feature representation capabilities.

Recent advancements in optimizing DETR models have primarily followed two complementary directions: architectural refinements and attention mechanism improvements. Architectural optimizations like RT-DETR [2] tackle computational bottlenecks in multi-scale processing by redesigning feature interaction paradigms, replacing the original costly simultaneous intra-scale and cross-scale operations with a more efficient hybrid encoder that strategically decouples and sequences these computations. Meanwhile, attention mechanism enhancements such as DeformableDETR [3]'s sparse query sampling and SparseDETR [4]'s dynamic token pruning directly attack the quadratic complexity of vanilla attention by introducing sparsity-either through learned deformable sampling locations or importance-based token selection. Complementing these query-side strategies, Focus DETR [5] tackles the attention mechanism's computational burden by shortening the encoder's query token sequence, employing a score-criticism network to selectively update only the most informative encoder elements [6]. While architectural changes focus on reorganizing computational pathways at the system level, attention modifications operate at the fundamental operator level; yet both approaches synergistically address different facets of the same efficiency challenge, with architectural innovations creating better-structured feature representations that subsequently enable more effective sparse attention mechanisms to operate. Current limitations persist as these optimizations remain largely structural, leaving opportunities for complementary parameter-level compression techniques to further eliminate redundancies in already optimized architectures [7].

Regarding the attention mechanism in the detection transformer, it facilitates the global context exchange between query tokens and key-value pairs. In the encoder, the query represents the entire feature map, while in the decoder, it is a learnable object query. The key and value correspond to the flattened multi-scale feature map tokens, including background tokens that contribute minimally to detection accuracy but consume significant computational resources. Existing methods

*Corresponding author: Zhe Wang (wangzhe@ecust.edu.cn).

This work was supported by the Natural Science Foundation of China under Grant No. 62476087 and 62306115, Shanghai Municipal Education Commission's Initiative on Artificial Intelligence-Driven Reform of Scientific Research Paradigms and Empowerment of Discipline Leapfrogging, and the Natural Science Foundation of Shanghai under the 2024 Shanghai Action Plan for Science, Technology and Innovation (No. 24ZR1416800).



(a) VOC0712 (b) MiniCOCO2017

Fig. 1. mAP compared with other lightweight DETR models on the VOC0712 and MiniCOCO2017 datasets.

[4], [5], [8] often focus solely on query token compression, overlooking the abundant background key and value tokens in the encoder and decoder. Moreover, the decoder layers vary in their need for location information granularity, with later layers requiring finer feature maps for bounding box refinement. However, many DETR models use uniform spatial feature maps in the decoder, leading to computational redundancy from flattened feature map tokens.

To tackle these issues, we introduce a novel weight-sharing softmask pruning strategy and spatial reduction token compression techniques. These techniques effectively minimize model parameters and computations while preserving detection accuracy. Specifically, our weight-sharing softmask pruning strategy is designed to differentially learn and prune the structures of MLP, Attention, and convolution blocks within the DETR model. During network training, it simultaneously learns the softmask value for each block, assigning an importance score based on which redundant, low-scoring structures are eliminated while preserving crucial, high-scoring ones. In the realm of token compression, we propose a spatial reduction token compression scheme that reduces the number of tokens participating in attention calculations by reducing feature maps which will be flatted into key and value token sequence in both encoder and decoder stages. This diminishes computational overhead in the sequence length dimension and can be seamlessly integrated with multi-head self-attention and multi-scale deformable attention mechanisms.

We summarize our contributions as follows:

- We propose a tailored weight-sharing softmask pruning strategy for DETR, enabling differential learning during optimization. This softmask evaluates parameter importance, guiding selective pruning of less critical structures. The strategy can flexibly control the degree of pruning and maintain the detection accuracy of the model.
- We design a spatial reduction token compression scheme to reduce key and value tokens sequence in attention, by reducing feature maps in both encoder and decoder stages, lowering computational cost. It integrates with multi-head self-attention and deformable attention, compressing tokens while maintaining detection accuracy.
- We apply the weight-sharing softmask pruning strategy and the spatial reduction token compression scheme to the DETR model, and propose a new detection transformer called the PR-DETR model. Experimental results on the *VOC0712* and *MiniCOCO2017* datasets show that the PR-DETR model outperforms the existing lightweight DETR model in terms of parameters and FLOPs, while maintaining a mAP comparable to the existing model.

The source code is publicly available at https://github.com/zhangmj86/pr-detr.git.

II. Related Work

A. Detection Transformer Model

Recently, a transformer-based end-to-end object detection model called DETR [1] has been proposed, which has high precision performance on object detection tasks. Although the design of DETR is good performance, it also has its own problems. It requires longer training time for convergence, and the performance in detecting small objects is relatively low. The self-attention's and cross-attention's calculation in the transformer encoder is a secondary calculation of the number of pixels within feature map.

Several methods have been proposed to accelerate training convergence and improve detection accuracy. For instance, DAB-DETR [9] introduces $4D$ anchor-box queries to improve interpretability, accelerate convergence, and modulate attention maps. DN-DETR [10] introduces a denoising training mechanism by injecting noisy boxes and labels during training, thereby accelerating convergence and improving robustness. DINO [11] further enhances denoising training to speed up optimization. GroupDETR [12] adopts a grouped one-vs-all label assignment strategy to alleviate the limited supervision in one-to-one matching, thereby facilitating faster convergence.

There are also some network mechanisms dedicated to lightweight DETR, which provide many reference methods for the practical application of DETR in the industry. DETR's standard Transformer, a multi-block stacking sequence modeling model, has massive parameters and quadratic computational complexity in its attention mechanisms. To address these problems, DeformableDETR [3] proposes deformable attention in decoder to reduce the computational cost associated with multi-scale attention. To leverage the advantages of Deformable-DETR and DN-DETR, researchers combined the denoising training mechanism of DN-DETR with the deformable attention of DeformableDETR, thus giving rise to the dn-DeformableDETR model. However, the computational cost of multi-scale feature map in DeformableDETR is still high. SparseDETR [4] proposes to use the foreground score predict network to sparse and update the queries of high importance in the encoder to reduce the number of queries. FocusDETR [5] proposes a top-down foreground token selector to allow the high-level feature map to guide the low-level feature map, which combined with multi-class score predictor to get category fine grain foreground tokens. LiteDETR [8] efficiently recalculates by staggering encoder updates of high-level and low-level feature maps, where the former queries the latter. RT-DETR [2] proposes an efficient multi-scale feature fusion strategy by optimizing the model structure and computational

flow, which simplifies the design of object queries, reduces the model complexity and computation.

B. Transformer Pruning

DETR variants accelerate inference via more efficient attention, token compression, and streamlined architectures. Yet the Transformer backbone still contains redundant parameters. The Lottery Ticket Hypothesis [13] suggests that a pruned subnetwork can be retrained to match the full model's performance, often with fewer training iterations.

Inspired by the Lottery Ticket Hypothesis, recent work has explored pruning Transformer architectures. X-Pruner [14] performs explainable structured pruning with learnable masks that estimate each component's contribution to class prediction. ViT-Slim [15] uses sparse masks for joint slimming, searching layer- and module-wise dimensions in a continuous space. Fast-Ptq-Trans [16] proposes a post-training pipeline that searches and refines masks to obtain a pruned subnetwork without retraining. While these pruning methods reduce model size, they are generally designed for classification backbones and do not consider the encoder-decoder structure and attention patterns specific to detection transformers.

C. Token Compression

The Transformer [17], [18] incurs quadratic self-attention cost with respect to token length, limiting large-scale deployment. Token-compression methods reduce this cost by merging or pruning tokens: Tome [19] merges similar tokens via bipartite soft matching, while Diffrate [20] learns an adaptive compression ratio (via reparameterization) and unifies token pruning with token merging. Pyramid-based ViTs such as PVT [21] progressively shrink token sequences and use convolutional spatial-reduction attention. Most token compression approaches target Vision Transformers for classification and focus on merging or pruning tokens, whereas detection transformers additionally suffer from abundant background key-value tokens that are rarely explored.

Taken together, prior work improves DETR efficiency mainly through architectural redesign and operator-level sparsification, while generic pruning and token compression methods remain tailored to classification backbones, overlooking detection-specific encoder–decoder structures. We address these gaps from two complementary angles: (i) detection-aware structured pruning via an end-to-end optimized weight-sharing softmask, and (ii) spatial reduction of key–value tokens to lower attention complexity beyond query-side compression. Both components are compatible with MHSA and MSDA, yielding consistent reductions in parameters and FLOPs while preserving competitive accuracy.

III. Method

A. Overview of PR-DETR

Fig. 2 shows the overall architecture of the proposed PR-DETR object detection model, which consists of a backbone network, an encoder, and a cascade decoder. The backbone extracts multi-scale feature pyramids, and the highest-level

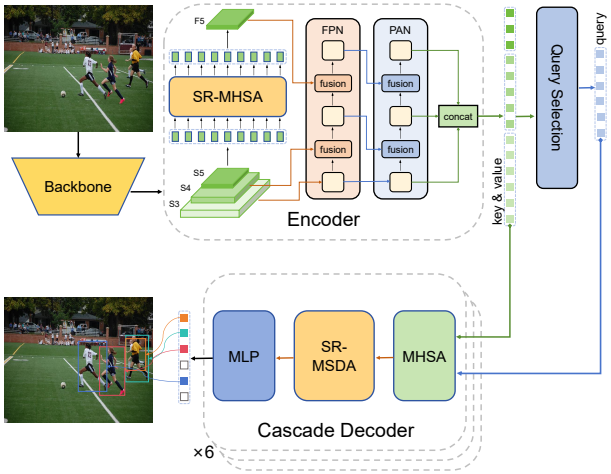


Fig. 2. Overview of our proposed PR-DETR model. It includes backbone, encoder and decoder. Backbone generates a multi-scale feature pyramid. Encoder selects the 5th high-level semantic feature map for global context learning of feature points. {s3, s4, f5} are flattened into key and value sequences, the query selector selects top-k object queries with highest scores. The cascade decoder computes the cross-attention of the query with the key and value sequences to obtain bounding boxes of object detection.

semantic feature map is fed into the encoder to facilitate global context modeling. The F5 feature map, together with S3 and S4, is then processed by FPN and PAN to perform multi-scale feature fusion and enhance classification and localization. The decoder selects the top-k tokens with the highest scores from multi-scale feature maps as object queries, and predicts bounding boxes and class labels via self-attention, cross-attention, and MLP layers. In the training process, subsequent decoder layers achieve detailed gradient optimization by learning the residual error between the previous layer's prediction and the actual target, and the final decoder layer can predict more accurate target positions and classification confidence.

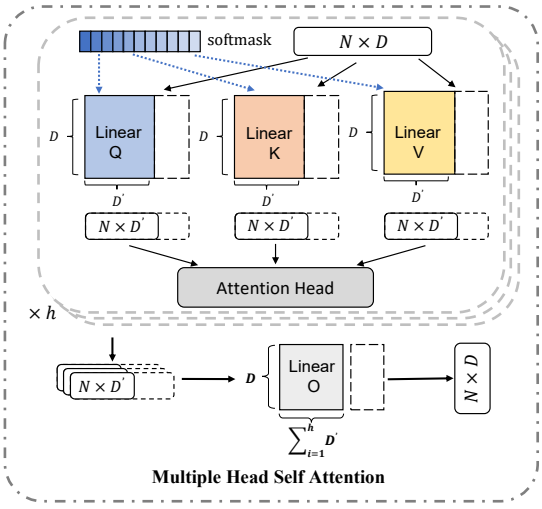
B. Weight-sharing Softmask Pruning Strategy

The lottery ticket hypothesis suggests that a neural network contains a subnetwork that, when trained independently, can match the original's performance with fewer iterations. In the transformer component of the DETR model, the parameter structure of the base layer presents a certain redundancy. Traditional mask-based pruning methods usually define the mask of the candidate layer for each base network layer within the predefined search space, and use an appropriate search algorithm to explore the parameters of the network architecture. However, these methods suffer from a longer search time led by a huge search space and higher GPU memory consumption led by amount of candidate layer.

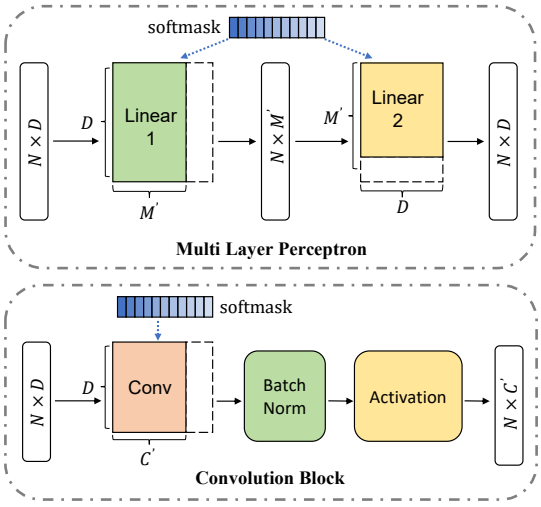
To address these issues, We propose direct channel pruning by sharing mask weights across all layers in the base block.

a) Weight-sharing Softmask: Let $w \in \mathbb{R}^{D_{in} \times D_{out}}$ represent the prunable parameters of a base network block (e.g., MLP, MHA, Conv), and $z \in \mathbb{R}^{1 \times D_{out}}$ be the softmask. The inference of the block is given by:

$$y = (xw) \odot z \tag{1}$$



(a) MHSA



(b) MLP and ConvBlock

Fig. 3. Our proposed weight-sharing softmask pruning strategy with MHSA, MLP and ConvBlock. Where N is the length of the token, D is the input dimension size of the token, and D', C', and M' are the dimensions size after pruning. h is the number of heads of MHSA.

where $\odot$ is element-wise multiplication and xw is the original output tensor.

If the preceding block is pruned, the input channels of the current block are pruned along with the output channels of the preceding block. In a residual structure with identical input and output dimensions for all residual paths, a single softmask z_r suffices for one residual path, and all paths share z_r.

As in Fig. 3 and Fig. 4, in the MHA mechanism, the linear projections such as *query*, *key*, *value*, and *output* share z. All linear projection layers in MLP share a z. For convolution blocks, a unique z is defined, and parent blocks (e.g., RepVGG [22]) share z across residual paths of all convolution blocks within the residual structure.

b) One Stage Search: When a differentiable softmask is integrated into each prunable base block, the network requires a single training iteration from scratch. The loss function for the differentiable learning softmask sub-neural architecture search within the object detection model is given by:

$$\min_{W,Z} \ \mathcal{J}(W,Z) = \mathcal{L}(f(X;W,Z),Y) + \lambda\|Z\|_1, \qquad (2)$$

Here, X and Y denote the input images and their corresponding ground-truth annotations, respectively. $f(\cdot;W,Z)$ denotes the detection network parameterized by the standard weights W and the learnable softmasks Z (shared across prunable structures in MHA/MLP/Conv/MSDA blocks), and $\mathcal{L}(\cdot,\cdot)$ is the detection loss (e.g., classification and box regression). The element-wise operator $\odot$ indicates that Z acts as a continuous gating mask on the associated prunable parameters/activations. λ is a regularization coefficient and the ℓ_1 penalty $\|Z\|_1$ encourages sparsity in Z, thereby promoting structured pruning by suppressing low-importance components while preserving high-importance ones.

c) Pruning and Fine-tuning: We denote the set of block types as $\mathcal{B} = \{\text{MHA}, \text{MLP}, \text{ConvBlock}, \text{MSDA}\}$. For $i \in \mathcal{B}$, let $\{S_i\}$ be the softmasks of the i-th type of block in the model. Upon completion of the network architecture search, we flatten and sort each S_i in descending order, denoted as $v_i = \text{sort}(\text{flatten}(S_i))$. Given a retention budget $\alpha = 0.7$, we determine the threshold t_i for the i-th type of block by $t_i = v_i[\lfloor \alpha \cdot |v_i| \rfloor]$. The conversion from softmask to hard mask H_i is given by:

$$H_{ij} = \begin{cases} 1, & \text{if } v_{ij} \geq t_i \\ 0, & \text{if } v_{ij} < t_i \end{cases} \qquad (3)$$

where v_{ij} is the j-th element of v_i.

For each block with parameter w, we prune the channel structures where the corresponding $H_{ij} = 0$, yielding a new block parameter $\hat{w}$. After pruning, all softmasks are removed from the network, and the forward pass of the pruned model is $Y = \hat{W} \cdot X$.

Since pruning alters the model's parameter space, retraining of the pruned model is necessary to achieve the original model's accuracy within the smaller pruned model. Given that pruning retains important structural parameters and the parameter distribution among them is valuable, the retraining process should utilize the same number of epochs as the original model, i.e., performance$(\hat{W}) \approx$ performance(W) after a full retraining process.

In the search phase, by implementing the weight-sharing softmask, which enables the assignment of a channel-attention weight to each basic block, our method can train a model that outperforms the baseline model in terms of mAP. The softmask acts as a continuous gating coefficient that progressively suppresses unimportant channels during training while preserving critical ones. Importance scores are jointly optimized with the detection objective through L1 regularization, enabling data-driven and architecture-adaptive pruning without handcrafted criteria.

C. Spatial Reduction Token Compression Scheme

As in Fig. 5, is our proposed spatial reduction token compression scheme with MSDA and MSDA. Let N_k denote the length of the key token sequence, N_q denote the length of the query token sequence, and C denote the dimension of the token vector. The computational complexity of the self-attention mechanism in the encoder and decoder phases is $O(N_k^2)$. In the decoder phase with the standard multi-headed cross-attention mechanism, the computational complexity is $O(N_qC^2 + N_k^2 + N_qN_kC)$. Given $N_k = hw$, where h and w represent the height and width of the feature map, large N_k leads to a computational complexity approaching $O(N_k^2)$. Therefore, we propose a spatial reduction token compression scheme that reduces the spatial resolution of multi-scale feature maps to balance accuracy and computational cost. Let $feat \in \mathbb{R}^{h \times w}$ be the input feature. First, we apply the adaptive global average pooling operation, denoted as AAP, with a fixed reduction rate r, which can be expressed as:

$$\hat{feat} = \text{AAP}_r(feat) \tag{4}$$

The AAP aggregates the spatial information of the feature map, reducing its spatial dimensions based on the specified reduction rate r. This operation preserves salient features while significantly shortening token sequences.

After performing the AAP, we flatten the $\hat{feat}$ into $1D$ vectors $\hat{k} \in \mathbb{R}^{hw}$ and $\hat{v} \in \mathbb{R}^{hw}$, which can be represented as follows:

$$\hat{k} = \text{Flat}(\hat{feat}), \hat{v} = \text{Flat}(\hat{feat}) \tag{5}$$

Here, Flat$(\cdot)$ is the flatten operation that converts the $2D$ feature map into a $1D$ vector.

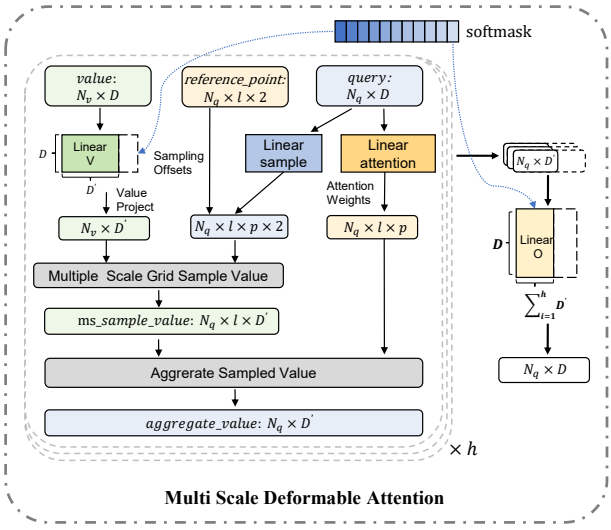


Fig. 4. Our proposed weight-sharing softmask pruning strategy with MSDA. Where N_q is the length of the query tokens, N_v is the length of value tokens, l is the number of feature maps, and p is the number of reference points of one query token.

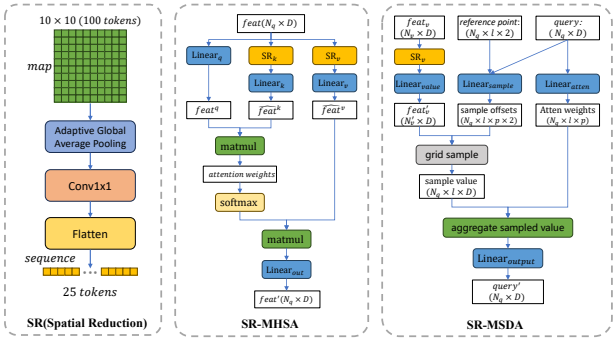


Fig. 5. Our proposed spatial reduction token compression scheme with MHSA and MSDA.

Finally, both $\hat{k}$ and $\hat{v}$ pass through a Conv$_{1\times1}$ layer. This Conv$_{1\times1}$ layer serves as a transformation layer that can adjust the feature representation, which can be described by:

$$\hat{k} = \text{Conv}_{1\times1}(\hat{k}), \hat{v} = \text{Conv}_{1\times1}(\hat{v}) \tag{6}$$

The Conv$_{1\times1}$ layer can learn to map the flattened features to a more suitable representation, potentially enhancing the expressiveness of the features for subsequent processing steps.

We combine the proposed spatial reduction token compression scheme with the standard multi-head attention to obtain a spatial reduction multi-head self-attention called SR-MHSA, which can be described as follows:

$$\hat{k}, \hat{v} = \text{SR}(feat) \tag{7}$$

$$head_j = \text{SelfAttention}(qw_j^q, \hat{k}w_j^k, \hat{v}w_j^v) \tag{8}$$

$$\text{SR-MHSA}(q, \hat{k}, \hat{v}) = \text{Concat}\left(head_j \mid j \in \{1, \ldots, h\}\right) \tag{9}$$

where q is the flatted token sequence of original $feat$, $\hat{k}$ and $\hat{v}$ are the flatted token sequences of $\hat{feat}$, j is the header index, h is the number of headers in multi-head attention, and SR$(\odot)$ denotes spatial reduction. The Concat$(\cdot)$ is the concatenation operation.

If the decoder uses Deformable attention [23], the computational complexity is $O(2N_qC^2 + \min(N_kC^2, N_qKC^2))$ with $K \leq 4$. Its complexity is near $O(N_qKC)$, but the *value* linear projection, using the original value token sequence, incurs redundancy. In the cascade decoder, higher layers need coarser high-level semantic maps for classification and coarse positioning, while lower layers need finer maps to refine bounding boxes. Our spatial reduction scheme integrates with multi-scale deformable attention to yield a spatial reduction multi-scale deformable attention mechanism. We set a layer-wise decreasing spatial reduction ratio for the cascade decoder. Our SR-MSDA is described by:

$$\hat{v} = \text{SR}(feat) \tag{10}$$

$$head_j = \text{DeformAtten}(qw_j^{offset}, qw_j^{atten}, \hat{v}w_j^v) \tag{11}$$

$$\text{SR-MSDA}(q, \hat{v}) = \text{Concat}(\text{head}_j \mid j \in \{1, \ldots, h\}) \tag{12}$$

where w_j^{offset} and w_j^{atten} are parameters of linear projection layers for learning sampling bias of object query and attention weight of sampled value sequence, respectively.

IV. Experiments

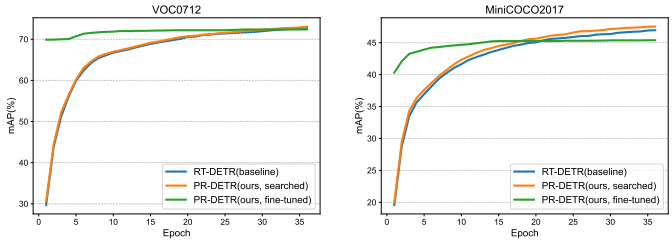
A. Experimental Setup

a) Dataset: In our experiments, we have utilized both the VOC0712 and MiniCOCO2017 datasets. We evaluate PR-DETR on VOC0712 and MiniCOCO2017 datasets for detection accuracy and computational efficiency. Specifically, the VOC0712 training set comprises 5011 images from Pascal VOC2007 [24] training portion and an additional 11,530 images from Pascal VOC2012 [24] training set. The corresponding test set for VOC0712 encompasses 4,952 images drawn from VOC2007's test set, spanning across 20 distinct categories. As for the MiniCOCO2017 dataset, in order to reduce the cost of model training and experiment time, it is constructed by randomly sampling one-third of the training images from the COCO2017 [25] dataset. The original COCO2017 training set contains 118,287 images, and the MiniCOCO2017 training set consists of 39,429 images obtained from this sampling process. Meanwhile, the test set of MiniCOCO2017 still uses the original 5,000 images from the val2017 subset of COCO2017. We evaluate detection performance using mAP on both VOC0712 and MiniCOCO2017 test sets. In addition to the mAP, we also report the number of parameters (Params), floating-point operations (FLOPs) and frames per second (FPS) to offer a more detailed analysis of the model's efficiency and complexity.

b) Implementation Details: Our experiments utilized a GPU server with an RTX 3090. We based our model on RT-DETR, using Resnet-50 [26] pre-trained on ImageNet [27] dataset as the backbone. To train our detector, we employed the *AdamW* optimizer, configuring it with a base learning rate of 0.0001, a weight decay factor of 0.0001, and a global gradient clipping norm set to 1. Additionally, we incorporated a linear warm-up phase spanning 2000 steps to stabilize the initial training. Exponential Moving Average (EMA) was applied with a decay rate of 0.9999 to further smoothen the model weights during evaluation.

B. Comparison with DETRs

As evident from Table I, our proposed PR-DETR model, under an 70% pruning budget on the MiniCOCO2017 dataset, achieves a notable reduction of 33.83% in parameters and 26.77% in FLOPs compared to the baseline RT-DETR. Although there is a modest decrease of 2.55% in mAP, PR-DETR demonstrates significant advantages over LiteDETR and FocusDETR in terms of model efficiency. Specifically, PR-DETR requires $7.861M$ fewer parameters than LiteDETR and $11.158M$ fewer than FocusDETR, while maintaining comparable mAP performance. Furthermore, in terms of FLOPs, PR-DETR achieves a reduction of 8.84 GFLOPs compared to FocusDETR, and only increase 2.09 GFLOPs compared to LiteDETR. On the VOC0712 dataset, our proposed method exhibits even more favorable performance, with a mere 0.89% decrease in mAP compared to the baseline RT-DETR, accompanied by a reduction of 37.230% in parameters and 39.23% in FLOPs.



(a) VOC0712 (b) MiniCOCO2017

Fig. 6. Tendency curves of mAP versus epoch for recovery accuracy of the fine-tuning phase model. It can be observed that our proposed PR-DETR, after pruning, is able to regain considerable accuracy within at the initial epoch on both the VOC0712 and MiniCOCO2017 datasets.

C. Ablation Study

a) Efficiency for Weight-sharing Softmask Pruning: Our weight-sharing softmask pruning strategy is selectively applied to the foundational blocks within the encoder and decoder stages, encompassing MHAs, MLPs, and Convs blocks with normalization and activation layers.

As evident from the ablation experimental results presented in Table II, on the VOC0712 dataset, the adoption of our weight-sharing softmask pruning approach leads to a relatively minor decrease of only 0.53% in mAP, while simultaneously achieving remarkable reductions of 49.30% in overall parameters (Params) and 39.23% in overall floating-point operations (FLOPs) compared to the baseline model. Specifically, within the encoder, we observe a significant reduction of 49.30% in Params and 37.92% in FLOPs. In contrast, the decoder shows more moderate reductions, with 17.52% in Params and 43.40% in FLOPs. Similarly, on the MiniCOCO2017 dataset, our weight-sharing softmask pruning method exhibits high efficiency. It results in substantial reductions of 46.11% in encoder Params, 26.27% in encoder FLOPs, 22.37% in decoder Params, and 25.46% in decoder FLOPs compared to the baseline model, accompanied by a modest decrease of 1.56% in mAP.

These results show that weight-sharing softmask pruning reduces parameters and FLOPs while maintaining competitive accuracy.

Furthermore, we analyzed the evolution of the mAP during the fine-tuning phase of the pruned model, as depicted in Fig. 6. Our proposed weight-sharing softmask pruning strategy showcases outstanding convergence characteristics.

b) Efficiency for Spatial Reduction: To ensure fairness and validate the effectiveness of the spatial reduction token compression scheme, we conducted ablation experiments by solely applying the spatial reduction scheme to the baseline model, as shown in Table II.

Our findings on the VOC0712 dataset indicate that, when employing spatial reduction alone, our method achieves an increase of approximately 0.55% in encoder Params and 4.51% in decoder Params compared to the baseline model. This increase in parameter count can be attributed to the $Conv_{1x1}$ layers defined within the adaptive global average

TABLE I
COMPARISON OF PR-DETR WITH STATE-OF-THE-ART DETR VARIANTS ON VOC0712 AND MINICOCO2017 DATASETS

Dataset	Model	Epoch	mAP%	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params(M)	FLOPs(G)	FPS
VOC0712	dn-DeformableDETR	50	66.03	87.06	72.01	27.03	47.37	73.25	17.80	70.52	36.60
	SparseDETR	50	56.67	82.85	61.93	17.51	39.59	63.85	12.33	24.27	44.41
	DINO	36	66.71	87.30	73.43	28.65	48.55	73.50	17.98	72.05	35.31
	FocusDETR	36	64.79	86.22	71.36	25.85	45.86	71.75	19.96	33.10	37.23
	LiteDETR	36	65.66	86.85	71.90	24.36	47.72	72.71	18.18	22.37	27.75
	RT-DETR(*baseline*)	36	**72.83**	91.43	79.70	36.70	57.23	78.93	19.30	33.53	**58.75**
	PR-DETR(*ours, searched*)	36	72.48	91.12	79.50	38.66	56.10	78.80	19.71	31.13	52.08
	PR-DETR(*ours, fine-tuned*)	36	71.94	90.84	78.81	37.31	55.93	78.26	**12.12**	**20.38**	52.42
MiniCOCO2017	dn-DeformableDETR	50	40.04	57.54	42.63	19.45	42.37	58.14	17.80	70.52	36.32
	SparseDETR	50	33.05	51.20	34.94	15.65	35.13	48.10	**12.55**	24.69	46.48
	DINO	36	40.59	58.32	43.29	21.30	42.89	57.34	18.09	72.21	35.02
	FocusDETR	36	38.54	56.54	41.11	19.72	40.70	54.60	20.07	33.50	37.25
	LiteDETR	36	39.56	57.32	42.29	19.69	42.29	56.60	18.22	22.57	27.73
	RT-DETR(*baseline*)	36	**46.95**	64.47	50.79	28.28	51.20	64.79	19.42	33.67	**58.38**
	PR-DETR(*ours, searched*)	36	45.92	63.43	49.49	26.29	50.59	64.24	19.83	31.27	52.27
	PR-DETR(*ours, fine-tuned*)	36	44.40	62.44	47.65	24.68	48.75	62.39	12.85	**24.65**	53.14

TABLE II
ABLATION STUDY ON EFFECTIVENESS OF WEIGHT-SHARING SOFTMASK PRUNING (SM) AND SPATIAL REDUCTION TOKEN COMPRESSION (SR) ON VOC0712 AND MINICOCO2017

Dataset	SM	SR	mAP(%)	Params(M)		FLOPs(G)		FPS
				encoder	decoder	encoder	decoder	
VOC0712			72.83	11.95	7.35	25.51	8.03	58.75
	✓		72.29	6.30	5.60	14.17	5.85	58.85
		✓	72.35	12.02	7.68	25.42	5.71	54.10
	✓	✓	71.94	6.06	6.06	15.84	4.54	52.42
MiniCOCO2017			46.95	11.96	7.47	25.51	8.16	58.38
	✓		45.39	6.44	5.80	18.81	6.08	57.90
		✓	46.42	12.02	7.80	25.42	5.85	53.98
	✓	✓	44.40	6.57	6.28	20.11	4.54	53.14

TABLE III
PRUNING BUDGET SENSITIVITY

Dataset	Budget	mAP	Enc&Dec		FPS
			Params(M)	FLOPs(G)	
VOC0712	50	70.65	8.46	16.28	61.77
	70	72.29	12.24	24.89	59.03
	80	72.66	14.54	28.46	57.98
	90	72.93	17.09	31.32	57.41
	100	73.07	19.42	33.67	56.93
MiniCOCO2017	50	42.33	7.91	12.63	59.61
	70	45.39	11.91	20.02	57.99
	80	46.03	14.36	24.83	57.33
	90	46.17	16.95	29.70	57.26
	100	47.49	19.30	33.53	56.59

pooling. However, we observe a substantial reduction in FLOPs for both the encoder and decoder, with decreases of approximately 0.34% and 28.84%, respectively. This shows that spatial reduction introduces minimal parameter overhead while substantially decreasing computational cost, especially in the cascaded decoder. Analogously, on the MiniCOCO2017 dataset, our approach leads to an increase of approximately 0.50% in encoder Params and 4.42% in decoder Params compared to the baseline model, accompanied by reductions of approximately 0.34% and 28.35% in FLOPs. Overall, these results indicate that weight-sharing softmask pruning primarily reduces model parameters, while spatial reduction mainly decreases FLOPs. Their combination provides complementary benefits, achieving the best trade-off between efficiency and detection accuracy.

D. Parameter Sensitivity Analysis

a) Budget for Softmask Pruning: As depicted in Table III, we systematically varied the pruning budgets from 50% to 90%, where the budget denotes the percentage of parameters retained, and presented the corresponding mAP as a function of these budgets. Our observations reveal a gradual decline in mAP as the pruning budget escalates, accompanied by a commensurate reduction in both Params and FLOPs. This underscores the flexibility of our method in tailoring the pruning budget to specific requirements, thereby striking an effective equilibrium between parameter efficiency, computational complexity, and model accuracy.

TABLE IV
IMPACT OF SPATIAL REDUCTION RATIO ON MODEL PERFORMANCE

Reduction Ratio		mAP	Params(M)		FLOPs(G)		FPS
encoder	decoder		encoder	decoder	encoder	decoder	
[0]	[0, 0, 0, 0, 0, 0]	46.95	11.95	7.47	25.51	8.16	58.38
[0]	[6, 4, 4, 2, 2, 1]	46.53	11.95	7.80	25.51	6.14	54.28
[2]	[6, 4, 4, 2, 2, 1]	46.40	12.02	7.80	25.42	6.14	53.68
[2]	[8, 6, 6, 4, 2, 1]	46.42	12.02	7.80	25.42	5.85	53.98
[2]	[4, 4, 4, 4, 4, 4]	45.20	12.02	7.87	25.42	5.29	53.34
[2]	[2, 2, 2, 2, 2, 2]	46.80	12.02	7.87	25.42	6.54	53.40

b) Reduction Ratio for Spatial Reduction: We performed experiments with varying reduction ratios and summarized the results in Table IV. We observed that as the reduction ratio decreased, the reduction level of the feature map also decreased, causing the length of the token sequence used in the attention mechanism to shorten, which in turn reduced FLOPs. However, this spatial reduction came at the cost of information loss, leading to a decrease in model accuracy.

In the decoder, a gradually decreasing reduction ratio outperformed a uniform one despite more parameters. This is due to our cascade refinement approach: early layers roughly position targets, using a higher reduction ratio for a smaller feature map. As layers progress, needing precise location information, the reduction ratio decreases.

V. Conclusion

In conclusion, our study presents an integrated approach combining weight-sharing softmask pruning and spatial reduction token compression to address the inefficiencies of the DETR framework. By analyzing the encoder–decoder architecture in popular DETR variants, we identify redundancies in both structural parameters and background tokens. Inspired by the lottery ticket hypothesis and neural architecture search, we design a weight-sharing softmask pruning strategy that simplifies the encoder and decoder, significantly reducing computational cost. Furthermore, the spatial reduction token compression specifically targets redundancy in background tokens caused by high-level semantic interactions, while adaptively accommodating the varying granularity requirements of location information across decoder layers. Integrated with Multi-Head Self-Attention (MHSA) and Multi-Scale Deformable Attention (MSDA) mechanisms, the approach achieves efficient token compression during both training and inference. Experiments on VOC0712 and MiniCOCO2017 validate the effectiveness of our method, demonstrating improved processing efficiency while offering new insights into accelerating DETR-based models.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagosyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[2] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 16 965–16 974.

[3] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[4] B. Roh, J. Shin, W. Shin, and S. Kim, "Sparse detr: Efficient end-to-end object detection with learnable sparsity," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[5] D. Zheng, W. Dong, H. Hu, X. Chen, and Y. Wang, "Less is more: Focus attention for efficient detr," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 6651–6660.

[6] D. Li, L. Ding, Z. Wang, and K. Zhao, "DepressInstruct: Instruction tuning of large speech-language models for depression detection," *Inf. Fusion*, vol. 130, p. 104077, 2026.

[7] M. Yang and Z. Wang, "Image synthesis under limited data: A survey and taxonomy," *Int. J. Comput. Vis. (IJCV)*, vol. 133, no. 6, pp. 3689–3726, 2025.

[8] F. Li, A. Zeng, S. Liu, H. Zhang, H. Li, L. Zhang, and L. M. Ni, "Lite detr: An interleaved multi-scale encoder for efficient detr," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 18 558–18 567.

[9] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[10] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 46, no. 4, pp. 2239–2251, 2024.

[11] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[12] Q. Chen, X. Chen, J. Wang, S. Zhang, K. Yao, H. Feng, J. Han, E. Ding, G. Zeng, and J. Wang, "Group detr: Fast detr training with group-wise one-to-many assignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 6610–6619.

[13] J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding sparse, trainable neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[14] L. Yu and W. Xiang, "X-pruner: Explainable pruning for vision transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 24 355–24 363.

[15] A. Chavan, Z. Shen, Z. Liu, Z. Liu, K.-T. Cheng, and E. P. Xing, "Vision transformer slimming: Multi-dimension searching in continuous optimization space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 4931–4941.

[16] W. Kwon, S. Kim, M. W. Mahoney, J. Hassoun, K. Keutzer, and A. Gholami, "A fast post-training pruning framework for transformers," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 24 101–24 116.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[18] H. Tian, Q. Zhou, Z. Wang, Q. Zhang, X. Xu, and Z. Fu, "Translating image into labels: End-to-end and scalable multi-label image classifier with language transformer," *Eng. Appl. Artif. Intell. (EAAI)*, vol. 162, p. 112670, 2025.

[19] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your ViT but faster," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[20] M. Chen, W. Shao, P. Xu, M. Lin, K. Zhang, F. Chao, R. Ji, Y. Qiao, and P. Luo, "Diffrate: Differentiable compression rate for efficient vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 17 118–17 128.

[21] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 548–558.

[22] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13 733–13 742.

[23] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 4794–4803.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis. (IJCV)*, vol. 88, no. 2, pp. 303–338, 2010.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis. (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.