Supplementary Material:
Siamese DETR

A. More Details of DETR

A.1. Multi-head Attention

Single-head Attention (SHA) **.** We start with the attention mechanism with single head. Given the key-value sequence $\boldsymbol{x}_{kv}$, query sequence $\boldsymbol{x}_q$, and linear projection of the attention head f_v, f_k, f_q, we can compute so-called query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$ embeddings:

$$\begin{aligned}\boldsymbol{q} &= f_q(\boldsymbol{x}_q + \boldsymbol{\phi}_q);\\ \boldsymbol{k} &= f_k(\boldsymbol{x}_{kv} + \boldsymbol{\phi}_p);\\ \boldsymbol{v} &= f_v(\boldsymbol{x}_{kv}),\end{aligned} \tag{1}$$

where $\boldsymbol{\phi}_p$ is the positional embedding for the key-value sequences, and $\boldsymbol{\phi}_q$ is the positional embedding for the query sequences. And the attention outputs $\widehat{q}$ are computed by the aggregation of weighted values:

$$\widehat{q} = \texttt{SHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = \sum_j \alpha_{i,j}\boldsymbol{v}_j, \tag{2}$$

where the attention weights is based on softmax of scaled dot products between i-th query and j-th key:

$$\alpha_{i,j} = \texttt{Softmax}(\frac{\boldsymbol{q}_i\boldsymbol{k}_j^T}{\sqrt{d_k}}), \tag{3}$$

where d_k is a scaling factor.

Multi-head Attention (MHA) **.** Through concatenating N single-head attentions followed by a projection $f_{\texttt{MHA}}$, we can compute the multi-head attention:

$$\begin{aligned}\widehat{q} &= \texttt{MHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v})\\ &= f_{\texttt{MHA}}\Big(\texttt{Concat}\big[\texttt{SHA}_0(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}), \ldots, \texttt{SHA}_N(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v})\big]\Big).\end{aligned} \tag{4}$$

Note that the output $\widehat{q}$ is the same size as the input query sequences $\boldsymbol{x}_q$.

A.2. Bipartite Matching

Following [1, 2], we apply the Hungarian algorithm [3] to match the N predictions $\widehat{y}$ with the ground truth y. The matching loss $\mathcal{H}$ is defined as:

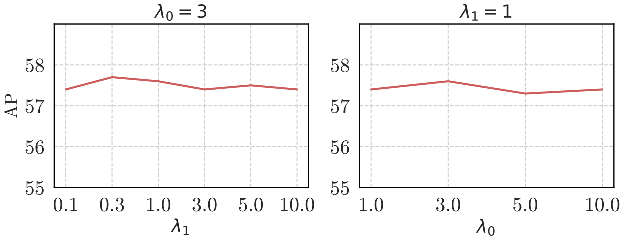

Figure 1. Sensitivity of Hyper-parameters in Siamese DETR.

$$\mathcal{H}(y, \widehat{y}) = \sum_{i=1}^{N}\Big[-\eta_0 \log \widehat{\boldsymbol{k}}_{\widehat{\sigma}(i)} + \mathbf{1}_{\{\mathrm{k_i}=1\}}\mathcal{L}_{box}(\boldsymbol{b}_i, \widehat{\boldsymbol{b}}_{\widehat{\sigma}(i)})\Big], \tag{5}$$

where $\boldsymbol{k}$ is the binary classification indicating whether each query is matched ($\boldsymbol{k}_i = 1$) or not ($\boldsymbol{k}_i = 0$), $\mathcal{L}_{box}$ is a combination of generalized IoU loss [6] and ℓ_1 loss, and $\widehat{\sigma}(i)$ is the index of prediction that matching with i-th ground truth optimally. The coefficients of binary classification η_0, generalized IoU loss η_1, and ℓ_1 loss η_2 in Equation 5 are set to 1, 2, 5 following [1], respectively.

B. More Ablations

Hyper-parameters. We follow [1] to set loss weight of $\mathcal{L}_{loc}$ (λ_2) to 1.0 in all setups and further ablate the λ_0 and λ_1 using Conditional DETR on COCO. Figure 1 illustrates the sensitivity of λ_0 and λ_1. It suggests that the transfer performance is robust to λ_0 and λ_1 variation. To yield the best performance, we set λ_0, λ_1 to 3, 10 on ImageNet, and λ_0, λ_1 to 0.3, 3 on COCO. For other novel datasets, a simple selection (*e.g.*, $\lambda_1 = 1.0, \lambda_2 = 1.0$) will be okay.

Data Efficiency. Self-supervised pretrained models not only achieve high performance when transferring to downstream tasks, but provide a better initialization when using limited data. To verify the data efficiency of Siamese DETR, we consider the transfer performance on the limited amount of downstream datasets. Specifically, we pretrain Conditional DETR on ImageNet and finetune it on 10%/30%/50%/70% PASCAL VOC datasets. All these

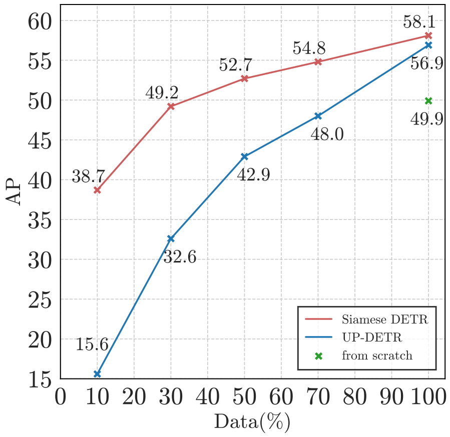

Figure 2. Data efficiency of Siamese DETR. We finetune Siamese DETR and UP-DETR using 10%/30%/50%/70% PASCAL VOC datasets.

Table 1. Ablations on downstream initialization. We initialize the Vanilla DETR using models pretrained by Siamese DETR, DETReg and UP-DETR without finetining. We report average recall with detecting top K objects, denoted as AR@K.

Method	AR@1	AR@10	AR@100
random	0.0	0.1	0.5
UP-DETR	9.5	17.9	24.0
DETReg	11.2	20.5	26.5
ours	12.4	23.0	30.7

Table 2. More DETR-like architecture. We pretrain the model using Siamese DETR on COCO for the 40/60 schedule, then finetune on full/10% PASCAL VOC dataset.

DETR	VOC		VOC 10%	
	DAB-DETR	DN-DETR	DAB-DETR	DN-DETR
from scratch	57.9	58.9	32.2	32.9
Siamese DETR	**62.2 (+4.3)**	**63.4 (+4.5)**	**41.8 (+9.6)**	**43.6 (+10.7)**

Table 3. Memory cost and iteration time during pretraining.

Method	GPU Mem.	Iteration time
Siamese DETR	7528 MB	0.4036 s/it
UP-DETR	7377 MB	0.3118 s/it
DETReg	8885 MB	0.3528 s/it

splits are selected randomly. As shown in Figure 2, Siamese DETR can achieve a similar (49.3 AP) performance with from scratch model (49.9) using only 30% of datasets. Moreover, Siamese DETR outperforms UP-DETR by a large margin in all splits.

Downstream initialization. To investigate the downstream initialization of Siamese DETR, we pretrain the Vanilla DETR on ImageNet and only finetune the box prediction and classifier head on PASCAL VOC while keeping the parameters of pretrained CNN backbone, encoder and decoder fixed. We report average recall with detecting top K objects, denoted as AR@K. As shown in Table 1, Siamese DETR outperforms its counterparts and random initialization.

More DETR-like architecture. We provide transfer results of Siamese DETR with more advanced DETR-like architecture, *i.e.*, DAB-Deformable-DETR with 300 queries [5] and DN-DAB-Deformable-DETR with 300 queries [4]. We follow the default setup in their origin paper. The results are shown in Table 2. Both DAB-DETR and DN-DETR can benefit from the initialization of Siamese DETR, verifying the generalization of Siamese DETR. Furthermore, Siamese DETR also leads a significant margin when using limited downstream datasets.

GPU Memory Cost and Iteration Time. Firstly, it is emphasised that none of three pre-training methods increase extra cost in GPU memory and time during downstream finetuning. We provide a quantitative comparison on GPU memory cost and iteration time during pretraining in Table 3. All three methods follow the same setup, *i.e.*, pretraining Conditional DETR (100 queries) on COCO using the same GPU. The batch size is set to 4 on each GPU. The input images are processed with the same augmentation.

Compared with UP-DETR on GPU memory cost, there is a slight increase in Siamese DETR because the parameters in Siamese DETR are all shared. Besides, performing multi-view learning and adding crop-level features does not bring too much time cost.

C. More Visualization

C.1. Convergence

Figure 3 and Figure 4 illustrate the convergence curves of models finetuned on COCO and PASCAL VOC, respectively. The model initialized by Siamese DETR converges faster and outperforms its counterparts by significant margins in all setups.

C.2. More Qualitative Results

We also provide more qualitative results of box predictions and corresponding attention maps when initializing the downstream model using Siamese DETR, UP-DETR, and DETReg without finetuning in Figure 5. The visualization results verify better transferability of Siamese DETR against its counterpart.

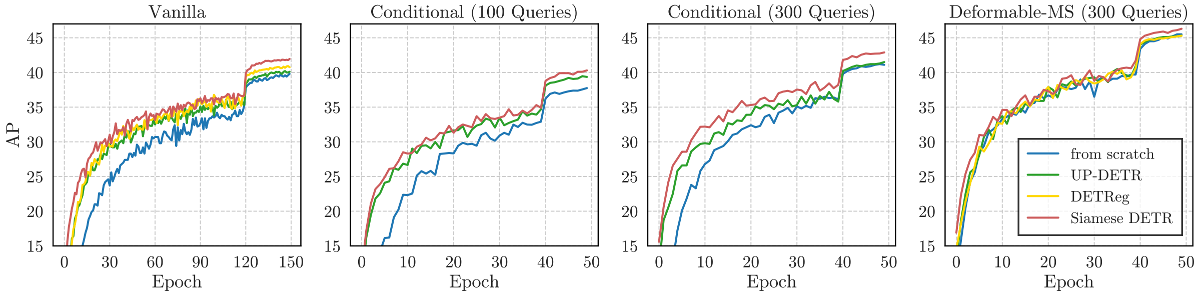

Figure 3. Illustration of convergence curves when finetuned on COCO.

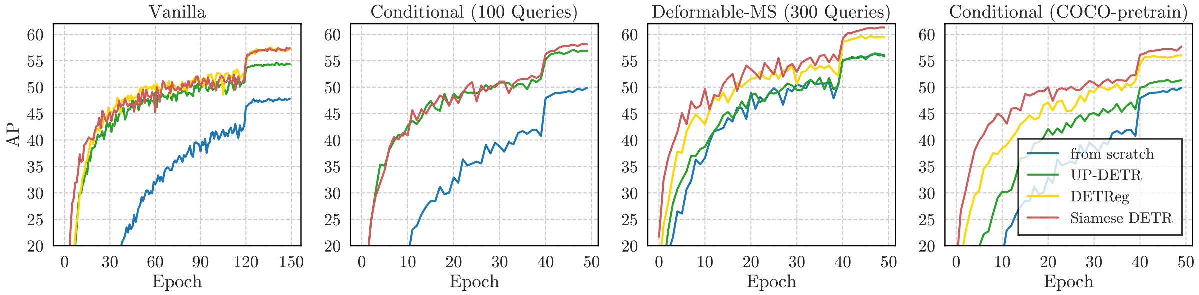

Figure 4. Illustration of convergence curves when finetuned on PASCAL VOC.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *eccv*, pages 213–229. Springer, 2020. 1

[2] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021. 1

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1

[4] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. 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, 2022. 2

[5] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2

[6] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019. 1

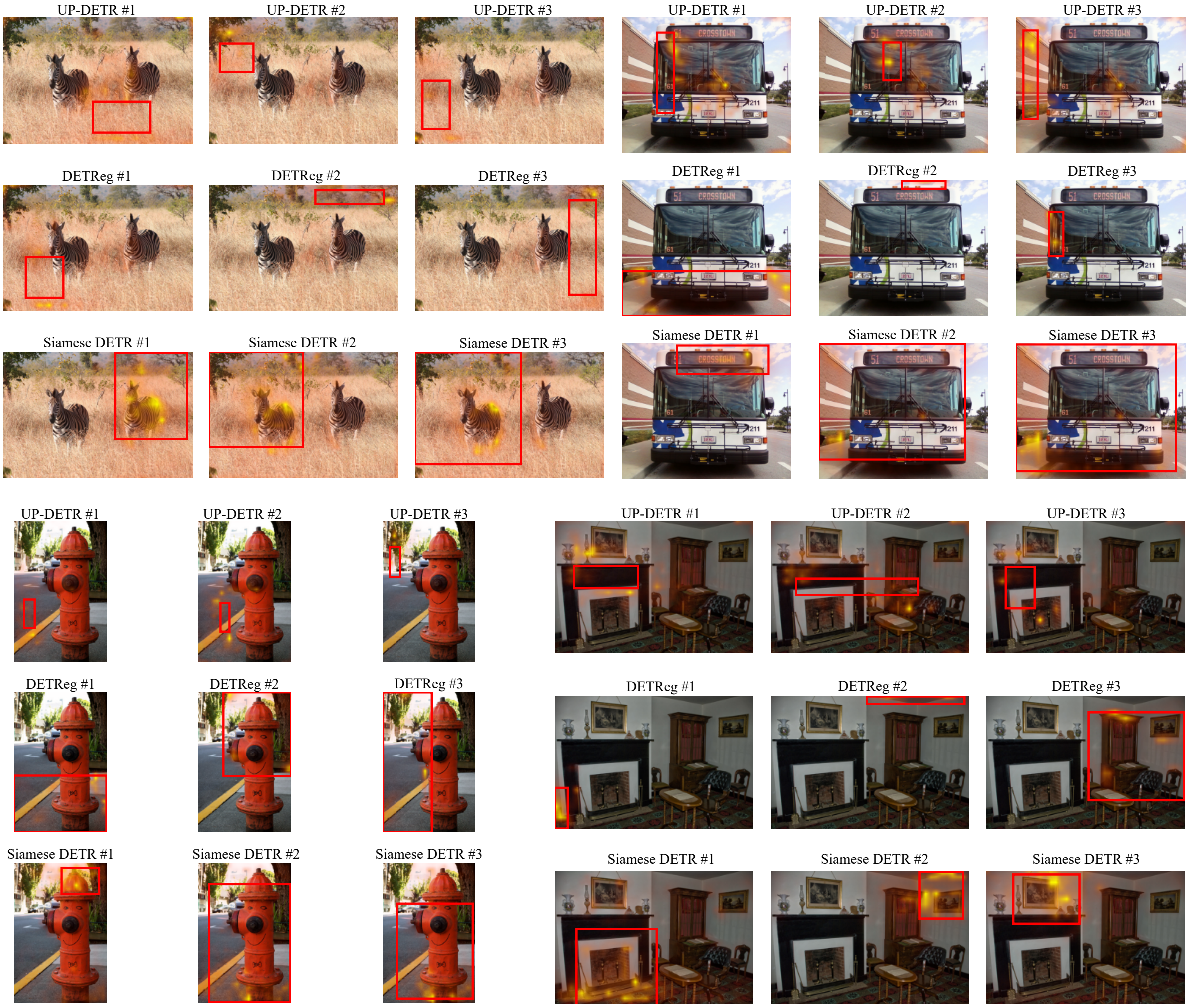

Figure 5. More visualization on box predictions and attention maps when initializing the downstream models using Siamese DETR, UP-DETR and DETReg without finetuning.