SIAMESE DETR

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent self-supervised methods are mainly designed for representation learning with the base model, *e.g.*, ResNets or ViTs. They cannot be easily transferred to DETR, with task-specific Transformer modules. In this work, we present **Siamese DETR**, a **Siamese** self-supervised pretraining approach for the Transformer architecture in **DETR**. We consider learning view-invariant and detection-oriented representations simultaneously through two complementary tasks, *i.e.*, localization and discrimination, in a novel multi-view learning framework. Two self-supervised pretext tasks are designed: (i) **Multi-View Region Detection** aims at learning to localize regions-of-interest between augmented views of the input, and (ii) **Multi-View Semantic Discrimination** attempts to improve object-level discrimination for each region. The proposed Siamese DETR achieves state-of-the-art transfer performance on COCO and PASCAL VOC detection using different DETR variants in all setups. Code will be made available.

1 INTRODUCTION

Object detection with Transformers (DETR) (Carion et al., 2020) combines convolutional neural networks (CNNs) and Transformer-based encoder-decoders, viewing object detection as an end-to-end set prediction problem. Despite its impressive performance, DETR and its variants still rely on large-scale, high-quality training data. It generally requires huge cost and effort to collect such massive well-annotated datasets, which can be prohibited in some privacy-sensitive applications such as medical imaging and video surveillance.

Recent progress in multi-view self-supervised representation learning (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; Caron et al., 2020a; Chen & He, 2021; Li et al., 2021; Grill et al., 2020) can potentially alleviate the appetite for labeled data in training DETR for object detection. However, these self-supervised learning approaches mainly focus on learning generalizable representations with base models, such as ResNets (He et al., 2016) and ViTs (Dosovitskiy et al., 2020). It is unclear how these approaches can be effectively extended to DETR with task-specific Transformers modules that are tailored for end-to-end object detection.

Designing self-supervised pretext tasks for pretraining the Transformers in DETR is a challenging and practical problem, demanding representations that could benefit object detection, beyond just learning generic representation. Several attempts have been made to address this issue. For example, UP-DETR (Dai et al., 2021) introduces an unsupervised pretext task based on random query patch detection, predicting bounding boxes of randomly-cropped query patches in the given image. Recent DETReg Bar et al. (2022) employs a pre-trained SwAV (Caron et al., 2020b) and offline Selective Search proposals (Uijlings et al., 2013) to provide pseudo labels for DETR pertaining. In general, both UP-DETR and DETReg follow a single-view pretraining paradigm (see Figure 1 **(a)**), without exploring the ability of learning transferable representations demonstrated in existing multi-view self-supervised approaches.

In this work, we are interested in investigating the effectiveness of multi-view self-supervised learning for DETR pre-training. To this end, we present a Siamese self-supervised pretraining approach for the Transformers in DETR, named Siamese DETR, with two proposed self-supervised pretext tasks dedicated to view-invariant representation learning. Specifically, given each unlabeled image, we follow (Wei et al., 2021; Bar et al., 2022) to obtain the offline object proposals and generate two augmented views guided by Intersection over Union (IoU) thresholds. As illustrated in Figure 1 **(b)**, by directly locating the query regions between augmented views and maximizing the discriminative

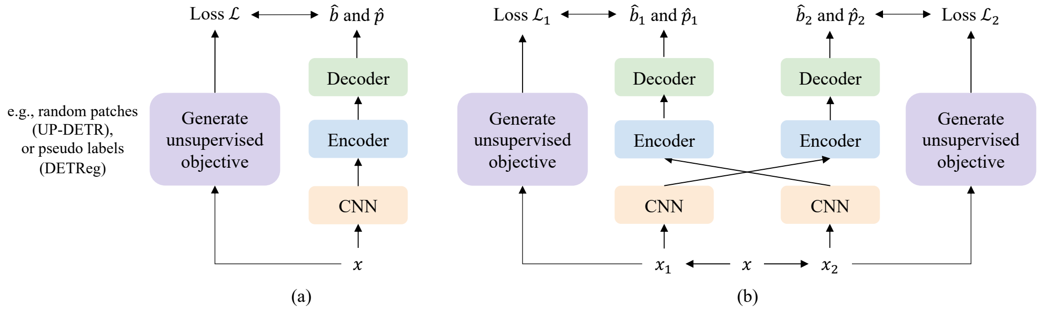
Loss $\mathcal{L}$
$\hat{b}$ and $\hat{p}$
Decoder
Encoder
CNN
e.g., random patches (UP-DETR), or pseudo labels (DETReg)
Generate unsupervised objective
x
(a)
Loss $\mathcal{L}_1$
$\hat{b}_1$ and $\hat{p}_1$
$\hat{b}_2$ and $\hat{p}_2$
Loss $\mathcal{L}_2$
Decoder
Decoder
Generate unsupervised objective
Encoder
Encoder
Generate unsupervised objective
CNN
CNN
x_1
x
x_2
(b)

Figure 1: Comparison between single-view and multi-view detection pretraining for DETR. **(a)** The single-view framework, *e.g.*, UP-DETR (Dai et al., 2021) and DETReg (Bar et al., 2022), perform self-supervised representation learning using unsupervised objectives generated on the single view, leading to a small information gain during pretraining. **(b)** The proposed multi-view Siamese DETR for DETR pretraining. Here, $\hat{b}$ and $\hat{p}$ denote box and semantic predictions, respectively.

information at both global and regional levels, Siamese DETR can learn view-invariant representations with localization and discrimination that are aligned with downstream object detection tasks during pretraining. Our contributions can be summarized as below:

- We propose a novel Siamese self-supervised approach for the Transformers in DETR, which jointly learns view-invariant representations with discrimination and localization. In particular, we contribute two new designs of self-supervised pretext tasks specialized for multi-view detection pretraining.
- Without bells and whistles, Siamese DETR outperforms UP-DETR (Dai et al., 2021) and DETReg (Bar et al., 2022) with multiple DETR variants, such as Conditional (Meng et al., 2021) and Deformable (Zhu et al., 2020) DETR, on the COCO and PASCAL VOC benchmarks, demonstrating the effectiveness and versatility of our designs.

2 Related Work

Object Detection with Transformers. DETR (Carion et al., 2020) integrates CNNs with Transformers (Vaswani et al., 2017), effectively eliminating the need for hand-crafted components, such as rule-based training target assignment, anchor generation, and non-maximum suppression. Several recent DETR variants (Zhu et al., 2020; Meng et al., 2021; Gao et al., 2021; Li et al., 2022; Zhang et al., 2022) have been proposed to improve the attention mechanism and bipartite matching in Transformers. For example, Zhu et al. (2020) only attend to a small set of key sampling points around a reference for faster convergence. Meng et al. (2021) learn a conditional spatial query from the decoder embedding for decoder multi-head cross-attention. Li et al. (2022) introduce a denoising pipeline to reduce the difficulty of bipartite matching.

In contrast, we explore another paradigm to improve the representations of Transformers for DETR via self-supervised pretraining. Dai et al. (2021) design a pretext task based on random query patch detection. Bar et al. (2022) train DETR using pseudo labels generated by pretrained SwAV (Caron et al., 2020b) and offline Selective Search proposals (Uijlings et al., 2013). While similarly following the existing *pre-train and fine-tune* paradigm, our work significantly differs from UP-DETR and DETReg. Specifically, (a) we perform view-invariant representation learning for the Transformers in DETR, and (b) to our knowledge, we make the first attempt to combine the Siamese network with the cross-attention mechanism in DETR in a self-supervised pretraining framework.

Self-supervised Pretraining. One of the main approaches for self-supervised learning is to compare different augmented views of the same data instances in the representation space. Some notable studies include that by Chen et al. (2020a), which presents a simple framework by removing the requirements of specialized architectures or memory banks. He et al. (2020) employ a momentum encoder with a dynamic dictionary look-up and retrieve more negative samples by using large dictionary sizes. Chen & He (2021) explore simple Siamese networks to learn meaningful representa-

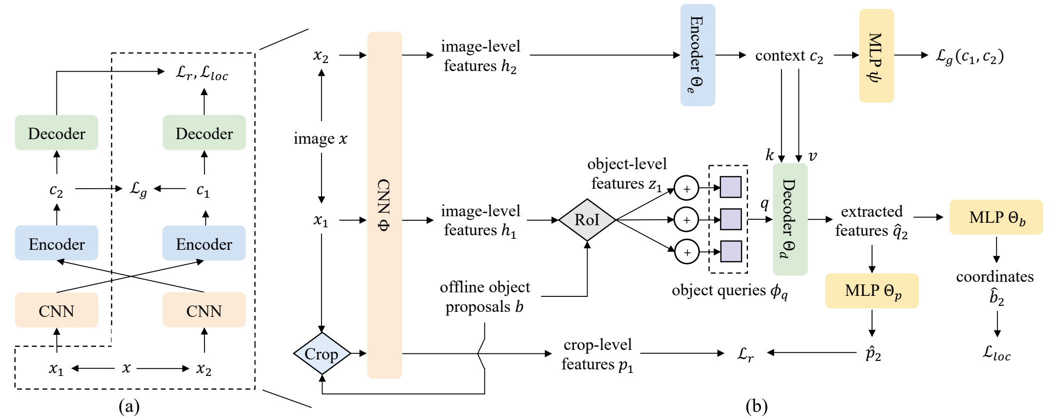

Figure 2: **(a)** Overall architecture of proposed Siamese DETR. **(b)** The forward process of one view in our symmetrical pipeline. We perform region detection ($\mathcal{L}_{loc}$) and semantic discrimination ($\mathcal{L}_g$ and $\mathcal{L}_r$) in a multi-view fashion. Given the conditional input of region features in one view, we aim at locating and discriminating their corresponding regions in another view.

tions with positive pairs only. Caron et al. (2020a) enforce consistency between cluster assignments produced for different augmentations of the same image. In addition, several attempts (Xie et al., 2021a; Zhao et al., 2021; Wang et al., 2021; Wei et al., 2021; Hénaff et al., 2021) have been made to learn detection-oriented representations directly using intrinsic cues, such as mask predictions (Zhao et al., 2021; Hénaff et al., 2021), offline region proposals (Wei et al., 2021; Xie et al., 2021b), and joint global-local partitions (Wang et al., 2021; Xie et al., 2021a). While different in the specific learning strategies, all these works focus on learning discriminative representations for base models, which are insufficient for transfer learning in DETR. In addition, the pretext tasks in these methods cannot be directly applied to the Transformers in DETR.

Siamese Networks. Siamese networks (Bromley et al., 1993) are weight-sharing neural networks and usually take two inputs for comparison, which are widely adopted in many applications, such as face verification (Taigman et al., 2014), person re-identification (Zheng et al., 2019), and one-shot learning (Koch et al., 2015). Recent advances in self-supervised learning (Chen et al., 2020b;a; Chen & He, 2021; Grill et al., 2020) are also built upon Siamese networks, motivating us to explore the Siamese architecture for pretraining the Transformers in DETR.

3 SIAMESE DETR

An overview of our Siamese DETR architecture is presented in Figure 2, which illustrates the main pipeline of our multi-view detection pretraining. We first revisit the DETR in Section 3.1. We then describe the view construction algorithm in Section 3.2 and the multi-view pretraining paradigm of Siamese DETR in Section 3.3, powered by two specially designed self-supervised pretext tasks for learning to detect objects.

3.1 REVISITING DETR

A typical DETR model consists of two modules: (i) a backbone model, *i.e.*, CNNs, for feature extraction, (ii) Transformers with encoder-decoders architecture for set prediction, built by stacking multi-head attentions (Vaswani et al., 2017). The backbone model extracts the image-level features $\boldsymbol{h} = \texttt{Backbone}(\boldsymbol{x})$ for a given image $\boldsymbol{x} \in \mathbb{R}^{3 \times H_0 \times W_0}$. Then, the Transformer encoder takes the image-level features $\boldsymbol{h}$ as inputs, encoding the image features as global context $\boldsymbol{c} \in \mathbb{R}^{C \times H_1 \times W_1}$:

$$\boldsymbol{c} = \texttt{Encoder}(\boldsymbol{h}). \tag{1}$$

Note that we omit the positional embedding ϕ_p in the subsequent description for clarity. The cross-mechanism is a general form of multi-head attention (MHA) in the Transformer decoder, which calculates the weighted sum $\widehat{\boldsymbol{q}} \in \mathbb{R}^{N \times C}$ between the flattened global context $\boldsymbol{c} \in \mathbb{R}^{H_1 W_1 \times C}$ and N

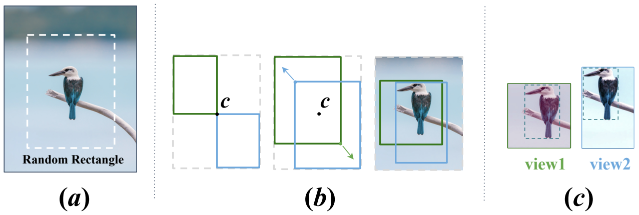

Figure 3: View construction using an IoU-constrained policy. ***(a)*** Generate a random rectangle within the image. ***(b)*** Generate two sub-rectangles on both sides of the center point c. Expand two rectangles along the diagonal while keeping the IoU larger than a threshold τ. Crop two sub-rectangles and apply augmentations. ***(c)*** Generate offline object proposals in the overlapping area.

object queries $\phi_q \in \mathbb{R}^{N\times C}$ for further box and categorical prediction.

$$\widehat{\boldsymbol{q}} = \texttt{CrossAtten}(\boldsymbol{c}, \boldsymbol{\phi}_q) = \texttt{MHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = \sum \texttt{Softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d_k}})\boldsymbol{v}. \tag{2}$$

The attention weights and summations are obtained by the query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$, which are the linear mapping of the context $\boldsymbol{c}$ and the object queries $\boldsymbol{\phi}_q$:

$$\boldsymbol{q} = f_q(\boldsymbol{\phi}_q); \quad \boldsymbol{k} = f_k(\boldsymbol{c}); \quad \boldsymbol{v} = f_v(\boldsymbol{c}). \tag{3}$$

Here f_q, f_k, f_v denote the projection for query, key and value in the cross-attention module. More details of DETR are provided in Appendix A.

In this work, we aim to pre-train the Transformers of DETR in a self-supervised way, extending the boundary of existing self-supervised pretraining. Motivated by recent advances in this paradigm (Chen et al., 2020b;a; Chen & He, 2021; Grill et al., 2020), we propose Siamese DETR, a Siamese multi-view self-supervised framework designed for Transformers in DETR, in which the model parameters and the learnable object queries in two DETRs are all shared. Following UP-DETR (Dai et al., 2021), Siamese DETR aims at learning representations with Transformers in self-supervised pretraining while keeping the backbone model frozen.

3.2 View Construction

We start with generating two views $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ for each unlabeled image $\boldsymbol{x}$, allowing the model to learn view-invariant object-level representations in self-supervised detection pretraining. As illustrated in Figure 3, we introduce an IoU-constrained policy to balance the shared information between two views. First, we generate a random rectangle within the image, which covers most content (50% to 100%). Then we use the center point of the rectangle as the anchor to create two sub-rectangles. By randomly expanding the sub-rectangles along the diagonal, we obtain two rectangles with the IoU larger than a threshold $\tau = 0.5$. Two rectangles are cropped from the image as the final two views $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. We further apply randomly and independently sampled transformations on two views $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ following the existing augmentation pipeline in Chen & He (2021). We also apply a box jitter processing following Wei et al. (2021) to encourage variance of scales and locations of object proposals across views.

The two augmented views $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ are visually distinct but share adequate semantic content. Following (Bar et al., 2022; Wei et al., 2021), we generate offline object proposals $\boldsymbol{b}$ using unsupervised EdgeBoxes (Zitnick & Dollár, 2014) [1] in the overlapping area between the two views and randomly select $n = 10$ corresponding object proposals in two views $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ from $\boldsymbol{b}$. These object proposals can provide proper objectness priors to learn object-level representations during pretraining.

3.3 Multi-view Detection Pretraining

Given a region extracted from one view, we aim to train the model to answer two questions: (1) *where is the corresponding region in another view?* (2) *what is the located region,* i.e., *does it*

[1] https://github.com/opencv/opencv_contrib. Apache-2.0 License.

semantically similar? In the following, we describe two pretext tasks designed in Siamese DETR: (1) learning to locate by **Multi-View Region Detection** and (2) learning to discriminate by **Multi-View Semantic Discrimination**.

Preparation. We take the augmented views $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ as inputs to the backbone model to obtain the image-level features: $\{\boldsymbol{h}_1, \boldsymbol{h}_2\} = \texttt{Backbone}(\{\boldsymbol{x}_1, \boldsymbol{x}_2\})$. Then, the object-level region features $\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ are extracted from the image-level features $\{\boldsymbol{h}_1, \boldsymbol{h}_2\}$ based on each object proposal $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ using RoIAlign (He et al., 2017):

$$\boldsymbol{z}_1 = \texttt{RoIAlign}(\boldsymbol{h}_1, \boldsymbol{b}_1); \;\; \boldsymbol{z}_2 = \texttt{RoIAlign}(\boldsymbol{h}_2, \boldsymbol{b}_2). \tag{4}$$

We also obtain the corresponding crop-level region features $\{\boldsymbol{p}_1, \boldsymbol{p}_2\} \in \mathbb{R}^{C \times H_3 \times W_3}$ by first cropping from the augmented views and then extraction:

$$\boldsymbol{p}_1 = \texttt{Backbone}(\texttt{Crop}(\boldsymbol{x}_1, \boldsymbol{b}_1)); \;\; \boldsymbol{p}_2 = \texttt{Backbone}(\texttt{Crop}(\boldsymbol{x}_2, \boldsymbol{b}_2)). \tag{5}$$

By default, both $\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ and $\{\boldsymbol{p}_1, \boldsymbol{p}_2\}$ are processed with global average pooling before further usage. Only the forward pass is involved in preparations, as we focus on pretraining the Transformers.

Multi-View Cross-Attention. We propose a Multi-View Cross-Attention (MVCA) mechanism that extends the cross-attention module in DETR for multi-view representation learning. With the introduced notions for two views $\{(\boldsymbol{c}_1, \boldsymbol{z}_1, \boldsymbol{p}_1, \boldsymbol{b}_1), (\boldsymbol{c}_2, \boldsymbol{z}_2, \boldsymbol{p}_2, \boldsymbol{b}_2)\}$, we formulate the cross-attention from view $\boldsymbol{x}_1$ to view $\boldsymbol{x}_2$ as follows:

$$\widehat{\boldsymbol{q}}_2 = \texttt{MVCA}(\boldsymbol{c}_2, \boldsymbol{\phi}_q, \boldsymbol{z}_1) = \texttt{MHA}(\boldsymbol{q}_1, \boldsymbol{k}_2, \boldsymbol{v}_2), \tag{6}$$

where the query $\boldsymbol{q}_1$, key $\boldsymbol{k}_2$ and value $\boldsymbol{v}_2$ are given by:

$$\boldsymbol{q}_1 = f_q(\boldsymbol{z}_1 + \boldsymbol{\phi}_q); \;\; \boldsymbol{k}_2 = f_k(\boldsymbol{c}_2); \;\; \boldsymbol{v}_2 = f_v(\boldsymbol{c}_2). \tag{7}$$

We add the region features $\boldsymbol{z}_1$ from view $\boldsymbol{x}_1$ to the object queries $\boldsymbol{\phi}_q$ so that with conditional input of region features $\boldsymbol{z}_1$, the object queries $\boldsymbol{\phi}_q$ can extract the relevant features $\widehat{\boldsymbol{q}}_2$ from the global context $\boldsymbol{c}_2$ of view $\boldsymbol{x}_2$. Here, $\widehat{\boldsymbol{q}}_2$ is supposed to be aggregated features on view $\boldsymbol{x}_2$ that are semantically consistent with the corresponding region features $\boldsymbol{z}_1$ on view $\boldsymbol{x}_1$.

Learning to locate: The MVCA mechanism allows us to conduct **Multi-View Region Detection** directly. Specifically, with the input of each region feature $\boldsymbol{z}_1$ from view $\boldsymbol{x}_1$ and its extracted feature $\widehat{\boldsymbol{q}}_2$, our goal is to locate the region in view $\boldsymbol{x}_2$ that is relative to the region feature $\boldsymbol{z}_1$. We apply a prediction head f_{box} for box prediction:

$$\widehat{\boldsymbol{b}}_2 = f_{box}(\widehat{\boldsymbol{q}}_2) \in \mathbb{R}^{N \times 4}. \tag{8}$$

After performing bipartite matching (Carion et al., 2020), we calculate the multi-view symmetrical localization loss as:

$$\mathcal{L}_{loc} = \ell_{box}(\widehat{\boldsymbol{b}}_2, \boldsymbol{b}_2) + \ell_{box}(\widehat{\boldsymbol{b}}_1, \boldsymbol{b}_1), \tag{9}$$

where ℓ_{box} is a combination of generalized IoU loss and ℓ_1 loss the same as Carion et al. (2020).

Learning to discriminate: Due to the unavailability of semantic label information, we propose **Multi-View Semantic Discrimination** to learn to discriminate at both global and regional levels. First, we apply a prediction head f_{sem} for further discriminative learning:

$$\widehat{\boldsymbol{p}}_2 = f_{sem}(\widehat{\boldsymbol{q}}_2) \in \mathbb{R}^{N \times C'}. \tag{10}$$

Considering that the context $\boldsymbol{c}$ of each view contains global contextual information, we maximize the similarity of the encoded context between two augmented views. Following Chen & He (2021), we apply a three-layer MLP (FC-BN-ReLU) and compute the global discrimination loss $\mathcal{L}_g$ symmetrically as:

$$\mathcal{L}_g = \mathcal{C}\big[\texttt{MLP}(\boldsymbol{c}_1), \texttt{detach}(\boldsymbol{c}_2)\big] + \mathcal{C}\big[\texttt{MLP}(\boldsymbol{c}_2), \texttt{detach}(\boldsymbol{c}_1)\big], \tag{11}$$

where $\mathcal{C}$ is the negative cosine similarity. In addition, due to the semantic consistency between the input region features $\boldsymbol{z}_1$ and the extracted features $\widehat{\boldsymbol{q}}_2$, we consider maximizing the semantic consistency for each region. Here, despite representing the same instances, the crop-level region features $\boldsymbol{p}_1$ can provide more discriminative information than object-level region features $\boldsymbol{z}_1$. It is because the object-level region features $\boldsymbol{z}_1$ are extracted from the image-level features (See Equation 4 and 5) and contain an aggregate of surrounding contexts with less discriminative information on themselves, especially for small regions. It motivates us to replace object-level region features $\boldsymbol{z}_1$ with crop-level region features $\boldsymbol{p}_1$ as the learning objectives. Besides, reconstruction with crop-level region features $\boldsymbol{p}_1$ can avoid potential degeneration problems in UP-DETR. Finally, we formulate the semantic consistency objective to improve the region discrimination as:

$$\mathcal{L}_r = \mathcal{D}(\widehat{\boldsymbol{p}}_2, \boldsymbol{p}_1) + \mathcal{D}(\widehat{\boldsymbol{p}}_1, \boldsymbol{p}_2), \tag{12}$$

where $\mathcal{D}$ is the normalized ℓ_2 distance.

Loss Function. Formally, the overall loss function for Siamese DETR is formulated as:

$$\mathcal{L} = \lambda_0 \mathcal{L}_r + \lambda_1 \mathcal{L}_g + \lambda_2 \mathcal{L}_{loc}, \tag{13}$$

where $\lambda_{0/1/2}$ are the loss weighting hyper-parameters.

4 Experiments

4.1 Implementation Details

Architecture. Siamese DETR consists of a frozen ResNet-50 backbone, pretrained by SwAV (Caron et al., 2020b) on ImageNet, and a Transformer with encoder-decoder architecture. Both the Transformer encoder and decoder are stacked with 6 layers of 256 dimensions and 8 attention heads. To verify the performance and generalization of our design, we compare Siamese DETR with a baseline model without pretraining (denoted as *from scratch*), UP-DETR (Dai et al., 2021), and DETReg (Bar et al., 2022). We use three DETR variants, *i.e.*, original DETR (Carion et al., 2020) (denoted as *Vanilla DETR*), Conditional DETR (Meng et al., 2021), and Deformable DETR (Single-Scale and Multi-Scale, denoted as Deform-SS and Deform-MS, respectively) (Zhu et al., 2020). For a fair comparison, the number of object queries is 100 in Vanilla DETR and Conditional DETR. We also use 300 queries in Conditional DETR and Deformable DETR following their default setup. We implement Siamese DETR based on the MMSelfSup codebase[2].

Evaluation Protocol. We follow the evaluation protocol in UP-DETR. Specifically, we first pretrain DETR variants on ImageNet (∼1.28 million images) (Deng et al., 2009) or COCO `train2017` (∼118k images) (Lin et al., 2014) separately. Then, the ImageNet-pretrained models are finetuned on COCO `train2017` or PASCAL VOC `trainval07+12` (∼16.5k images) (Everingham et al., 2010) separately, while COCO-pretrained models are finetuned on PASCAL VOC `trainval07+12`. We report COCO-style metrics, including AP, AP_{50}, AP_{75}, AP_s, AP_m, AP_l, in both COCO `val2017` and PASCAL VOC `test2007` benchmarks.

Pretraining. We use an AdamW (Loshchilov & Hutter, 2017) optimizer with a total batch size of 256 on ImageNet and 64 on COCO, a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. We adopt a full schedule of 60 epochs, and the learning rate decays at 40 epochs, denoted as the 40/60 schedule for brevity. Unless specified, UP-DETR uses random boxes, DETReg uses proposals generated by Selective Search (Uijlings et al., 2013), and Siamese DETR uses Edgeboxes in experiments.

Finetuning. For Vanilla DETR, we adopt 120/150 and 40/50 schedules in COCO and PASCAL VOC benchmarks. The initial learning rates of the Transformer and backbone are set to 1×10^{-4} and 5×10^{-5}. For the other two DETR variants, we report the result under the 40/50 schedule. The initial learning rates of the Transformer and backbone are set to $1 \times 10^{-4}/2 \times 10^{-4}$ and $5 \times 10^{-5}/2 \times 10^{-5}$ in Conditional/Deformable DETR, respectively. For fair comparison, the batch size in all setup is set to 32.

4.2 Main Results

COCO Object Detection. Table 1 shows the transfer results on COCO. Siamese DETR achieves the best performance using three different DETR variants on all setups. Especially for DETR variants like Deformable DETR of Multi-Scale, Siamese DETR boosts the model upon baseline more significantly than UP-DETR and DETReg, demonstrating the compatibility of our design with different DETR architectures.

[2] https://github.com/open-mmlab/mmselfsup. Apache-2.0 License.

Table 1: Comparisons of Siamese DETR with supervised/UP-DETR/DETReg in the COCO detection benchmark. The results of all models are achieved by officially-released repositories and pretrained models. Here “#epoch” denotes the number of epochs in downstream finetuning.

Method	Backbone	DETR	#query	#epoch	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
from scratch	Sup. R50	Vanilla	100	150	39.5	60.3	41.4	17.5	43.0	59.1
from scratch	SwAV R50	Vanilla	100	150	39.7	60.3	41.7	18.5	43.8	57.5
UP-DETR	SwAV R50	Vanilla	100	150	40.5	60.8	42.6	19.0	44.4	60.0
DETReg	SwAV R50	Vanilla	100	150	41.9	61.9	44.1	19.1	45.7	61.5
ours	SwAV R50	Vanilla	100	150	**42.0**	**62.7**	**44.5**	**19.4**	**46.0**	**62.1**
from scratch	SwAV R50	Conditional	100	50	37.7	59.6	39.2	17.1	41.7	56.3
UP-DETR	SwAV R50	Conditional	100	50	39.4	61.2	41.0	18.1	43.0	58.7
ours	SwAV R50	Conditional	100	50	**40.5**	**61.6**	**42.6**	**19.5**	**44.2**	**60.1**
from scratch	SwAV R50	Conditional	300	50	41.1	62.3	43.4	20.6	45.0	59.4
UP-DETR	SwAV R50	Conditional	300	50	41.5	63.2	43.6	21.3	45.4	60.2
ours	SwAV R50	Conditional	300	50	**43.0**	**64.2**	**45.6**	**22.0**	**47.2**	**61.8**
from scratch	SwAV R50	Deform-SS	300	50	40.3	60.9	42.9	20.1	44.8	57.2
UP-DETR	SwAV R50	Deform-SS	300	50	40.8	61.8	43.4	20.4	45.1	59.1
ours	SwAV R50	Deform-SS	300	50	**42.1**	**62.8**	**44.7**	**22.3**	**46.6**	**59.9**
from scratch	SwAV R50	Deform-MS	300	50	45.5	64.2	49.4	27.8	49.2	59.4
UP-DETR	SwAV R50	Deform-MS	300	50	45.3	64.5	49.6	26.0	49.2	59.9
DETReg	SwAV R50	Deform-MS	300	50	45.5	64.1	49.9	26.9	49.5	59.6
ours	SwAV R50	Deform-MS	300	50	**46.3**	**64.6**	**50.5**	**28.1**	**50.1**	**61.5**

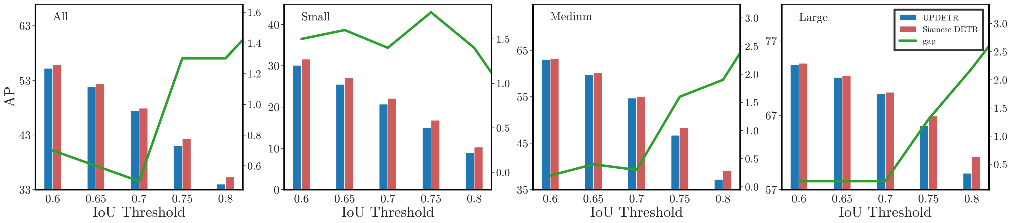

Figure 4: AP on COCO using different IoU Thresholds.

Besides, when adopting a stricter IoU threshold in metrics, *e.g.*, from AP_{50} to AP_{75}, Siamese DETR achieves a more considerable performance lead in most cases. It suggests the representations learned by Siamese DETR provide a stronger localization prior. We further illustrate AP metrics of Siamese DETR and UP-DETR using different IoU thresholds in Figure 4. Specifically, Siamese DETR performs well in localizing small objects and draws the gap against UP-DETR for medium and large objects when the IoU threshold is greater than 0.7.

PASCAL VOC Detection. Table 2 shows the transfer results on PASCAL VOC. Similar to conclusions on COCO, Siamese DETR achieves the best performance among all approaches on PASCAL VOC. We also report the result of COCO-pretrained models. Siamese DETR is 6.4 AP better than UP-DETR and 1.8 AP better than DETReg, which verifies the compatibility of Siamese DETR with different pretraining datasets, especially the scene-centric COCO.

4.3 Ablations

Effectiveness of Two Proposed Pretext Tasks. We train five Siamese DETR variants using Conditional DETR on ImageNet and finetune them on PASCAL VOC. Results are shown in Table 3. We treat UP-DETR as the baseline (56.9 AP on PASCAL VOC), which performs single-view patch detection with Transformer and reconstructs the decoder’s output with its input patches.

By extending single-view detection into a multi-view manner, (a) obtains a competitive result of 57.1 AP, suggesting that the view-invariant representations can perform better in downstream detection tasks. We further maximize the multi-view semantic consistency in terms of global and region discrimination, improving the transfer performance by 0.2 AP and 0.6 AP in (b) and (d), respectively. Global discrimination (b) brings smaller performance gains than region discrimination (d) in detection tasks, suggesting that it is impractical to directly apply existing instance discrimination pretext tasks (He et al., 2020; Koch et al., 2015) on Transformers of DETR in detection-oriented

Table 2: Comparisons of Siamese DETR with supervised/UP-DETR/DETReg in the PASCAL VOC detection benchmark. The results of all models are achieved by officially-released repositories and pretrained models. Here "#epoch" denotes the number of epochs in downstream finetuning.

Method	Backbone	DETR	Pretrain Dataset	#query	#epoch	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
from scratch	SwAV R50	Vanilla	-	100	50	28.5	47.5	29.4	1.3	7.4	40.3
UP-DETR	SwAV R50	Vanilla	ImageNet	100	50	50.0	73.5	53.4	6.5	29.5	64.1
DETReg	SwAV R50	Vanilla	ImageNet	100	50	53.8	76.5	57.3	**8.3**	**35.4**	67.5
ours	SwAV R50	Vanilla	ImageNet	100	50	**54.4**	**77.4**	**57.6**	7.7	35.0	**68.4**
from scratch	SwAV R50	Vanilla	-	100	150	47.8	73.8	50.9	5.4	27.6	61.4
UP-DETR	SwAV R50	Vanilla	ImageNet	100	150	54.4	78.1	58.6	10.5	35.8	67.5
DETReg	SwAV R50	Vanilla	ImageNet	100	150	57.0	79.7	61.6	11.5	**39.5**	70.2
ours	SwAV R50	Vanilla	ImageNet	100	150	**57.3**	**80.3**	**62.2**	**11.6**	39.3	**71.0**
from scratch	SwAV R50	Conditional	-	100	50	49.9	78.2	55.3	8.1	33.5	65.1.
UP-DETR	SwAV R50	Conditional	ImageNet	100	50	56.9	81.5	61.6	11.3	39.2	69.8
ours	SwAV R50	Conditional	ImageNet	100	50	**58.1**	**81.6**	**62.8**	**12.2**	**40.6**	**71.5**
from scratch	SwAV R50	Deform-SS	-	300	50	53.8	79.5	59.1	11.8	39.8	65.7
UP-DETR	SwAV R50	Deform-SS	ImageNet	300	50	54.0	79.3	58.8	11.4	38.6	66.1
ours	SwAV R50	Deform-SS	ImageNet	300	50	**58.0**	**81.8**	**64.0**	**14.0**	**43.3**	**70.3**
from scratch	SwAV R50	Deform-MS	-	300	50	56.1	80.7	61.9	17.4	42.7	66.4
UP-DETR	SwAV R50	Deform-MS	ImageNet	300	50	56.4	80.9	62.3	17.3	41.3	67.4
DETReg	SwAV R50	Deform-MS	ImageNet	300	50	59.7	82.0	66.4	18.2	46.4	70.4
ours	SwAV R50	Deform-MS	ImageNet	300	50	**61.2**	**82.9**	**67.7**	**19.3**	**47.1**	**72.2**
from scratch	SwAV R50	Conditional	-	100	50	49.9	78.2	55.3	8.1	33.5	65.1
UP-DETR	SwAV R50	Conditional	COCO	100	50	51.3	79.0	55.3	9.5	35.3	63.7
DETReg	SwAV R50	Conditional	COCO	100	50	55.9	80.0	61.6	**11.0**	39.3	68.5
ours	SwAV R50	Conditional	COCO	100	50	**57.7**	**80.9**	**62.5**	**11.0**	**40.4**	**70.9**

Table 3: Ablations on two proposed pretext tasks, *i.e.*, Multi-View Region Detection and Multi-View Semantic Discrimination. The notation "R-O" denotes region discrimination using object-level region features, "R-C" denotes region discrimination using crop-level region features, and "G" denotes global discrimination.

Method	Region Det.	Semantic Disc.	AP
UP-DETR	single-view	R-O	56.9
ours (a)	multi-view	-	57.1
ours (b)	multi-view	G	57.3
ours (c)	multi-view	R-O	57.3
ours (d)	multi-view	R-C	57.7
ours (e)	multi-view	R-C + G	**58.1**

Table 4: Comparisons of using different object proposals in Siamese DETR and its counterparts. The notation "A→B" denotes that the model is pretrained on dataset "A" and then finetuned on dataset "B".

Method	Dataset	Proposals	AP
UP-DETR	ImageNet→COCO	Random	39.4
ours	ImageNet→COCO	Random	**40.4**
UP-DETR	ImageNet→COCO	Edgeboxes	39.3
ours	ImageNet→COCO	Edgeboxes	**40.5**
UP-DETR	COCO→VOC	Random	51.3
DETReg	COCO→VOC	Random	51.9
ours	COCO→VOC	Random	**54.9**
DETReg	COCO→VOC	SelectiveSearch	55.9
ours	COCO→VOC	SelectiveSearch	**56.2**
UP-DETR	COCO→VOC	Edgeboxes	57.0
DETReg	COCO→VOC	Edgeboxes	56.3
ours	COCO→VOC	Edgeboxes	**57.7**

pretraining tasks. We also notice that (d) using crop-level region features achieves better performance than (c) using object-level region features, which verifies more discriminative information in crop-level region features. Finally, (e) with both Multi-View Region Detection and Multi-View Semantic Discrimination yields the best result of 58.1 AP.

Object Proposals. Edgeboxes (Zitnick & Dollár, 2014) in Siamese DETR provide objectness priors during pretraining. We attempt to replace it with boxes generated randomly or by Selective Search. All experiments are conducted using Conditional DETR. The results are shown in Table 4.

When pre-trained on object-centric datasets like ImageNet, which contains one single object in the center of the image, better objectness priors bring little improvement. In this case, our Siamese DETR still outperforms its counterparts by about 1.0 AP using random proposals or Edgeboxes.

When pre-trained on scene-centric datasets like COCO, which contains multiple objects in the image, great improvements are found in all methods after applying better objectness priors from proposals. Specifically, using random proposals, Siamese DETR outperforms UP-DETR by 3.6 AP and DETReg by 3.0 AP in the COCO→VOC setup, which suggests that Siamese DETR learns better detection-oriented representations without any objectness priors. When replaced with Selective Search and Edgeboxes, the performance gaps are alleviated. In this case, our Siamese DETR with Edgeboxes achieves the best performance among all setups.

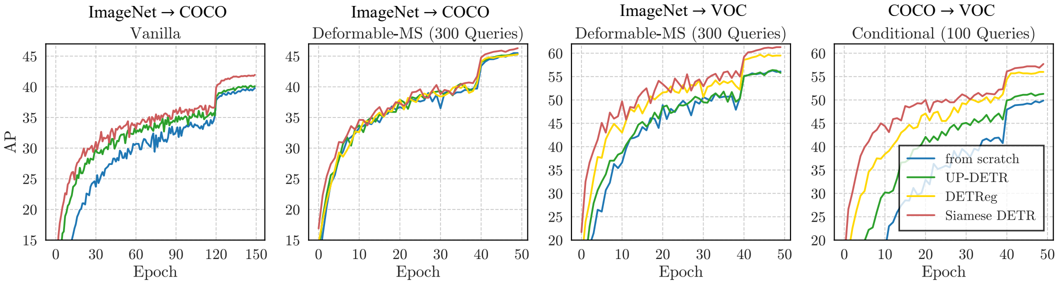

Figure 5: Illustration of convergence curves when fine-tuned on COCO or PASCAL VOC.

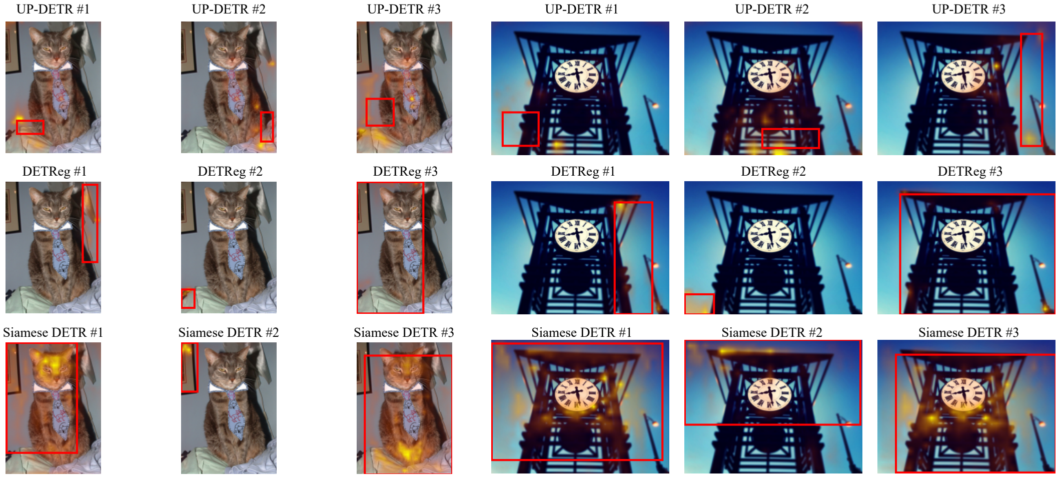

Figure 6: Visualization of box predictions and attention maps in downstream tasks. All these models (Vanilla DETRs) are initialized by Siamese DETR, UP-DETR, and DETReg without fine-tuning.

Convergence. The convergence curves of three DETR variants on downstream COCO and PASCAL VOC are illustrated in Figure 5. Compared with UP-DETR, DETReg, and the *from scratch* model, Siamese DETR converges faster and outperforms its counterparts by significant margins in all setups. More convergence curves are provided in Appendix C.1.

Visualization. We provide some qualitative results for further understanding the advantage of Siamese DETR. In downstream tasks, we use three Vanilla DETRs, initialized by Siamese DETR, UP-DETR, and DETReg pretraining models. Figure 6 illustrates their box predictions and corresponding attention maps in the decoder. Queries in Siamese DETR have stronger objectness priors, predicting more available box proposals overlapped with objects in the image. Meanwhile, benefitting from discriminative representations, cross-attention in Siamese DETR places more focus on the objects in the proposals. These qualitative results verify the transferability of Siamese DETR in downstream tasks. More visualizations are provided in Appendix C.2.

5 Conclusion and Limitations

In this paper, we propose Siamese DETR, a novel self-supervised pretraining method for DETR. With two newly-designed pretext tasks, we directly locate the query regions in a cross-view manner and maximize multi-view semantic consistency, learning localization and discrimination representations transfer to downstream detection tasks. Siamese DETR achieves better performance with three DETR variants in COCO and PASCAL VOC benchmark against its counterpart. Despite the great potential for pretraining DETR, Siamese DETR has a limitation in that it still relies on a pre-trained CNN, *e.g.*, SwAV, without integrating CNN and Transformer into a unified pretraining paradigm. In our future work, a more efficient schedule for the end-to-end DETR pretraining is desirable.

REFERENCES

Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14605–14615, 2022.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. *NeurIPS*, 6:737–744, 1993.

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

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020a.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pp. 15750–15758, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020b.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, pp. 3621–3630, October 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, pp. 10086–10096, October 2021.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICMLW*, volume 2. Lille, 2015.

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

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*, pp. 13619–13627, 2022.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *ICLR*, 2021.

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 *ECCV*, pp. 740–755. Springer, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv*, 2017.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pp. 3651–3660, 2021.

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*, pp. 658–666, 2019.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pp. 1701–1708, 2014.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pp. 3024–3033, 2021.

Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *arXiv*, 2021.

Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pp. 8392–8401, 2021a.

Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *arXiv*, 2021b.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *ICCV*, pp. 10160–10169, 2021.

Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. Re-identification with consistent attentive Siamese networks. In *CVPR*, pp. 5735–5744, 2019.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2020.

C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pp. 391–405. Springer, 2014.

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 parameters of the attention head $\boldsymbol{\theta} = \{\boldsymbol{\theta}_v, \boldsymbol{\theta}_k, \boldsymbol{\theta}_q\}$, we can compute so-called query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$ embeddings:

$$\boldsymbol{q} = f(\boldsymbol{x}_q + \boldsymbol{\phi}_q; \boldsymbol{\theta}_q); \boldsymbol{k} = f(\boldsymbol{x}_{kv} + \boldsymbol{\phi}_p; \boldsymbol{\theta}_k); \boldsymbol{v} = f(\boldsymbol{x}_{kv}; \boldsymbol{\theta}_v), \tag{14}$$

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:

$$\widehat{q} = \texttt{SHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}; \boldsymbol{\theta}) = \sum_j \alpha_{i,j} \boldsymbol{v}_j, \tag{15}$$

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

$$\alpha_{i,j} = \texttt{Softmax}(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^T}{\sqrt{d_k}}), \tag{16}$$

where d_k is a scaling factor.

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

$$\begin{aligned} \widehat{q} &= \texttt{MHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}; \boldsymbol{\theta}_{0\ldots N}, \boldsymbol{\theta}_{\texttt{MHA}}) \\ &= f\Big(\big[\texttt{SHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}; \boldsymbol{\theta}_0), \ldots, \texttt{SHA}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}; \boldsymbol{\theta}_N)\big]; \boldsymbol{\theta}_{\texttt{MHA}}\Big). \end{aligned} \tag{17}$$

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

A.2 Bipartite Matching

Following Carion et al. (2020); Dai et al. (2021), we apply the Hungarian algorithm Kuhn (1955) to match the N predictions $\widehat{y}$ with the ground truth y. The matching loss $\mathcal{H}$ is defined as:

$$\mathcal{H}(y, \widehat{y}) = \sum_{i=1}^{N} \Big[-\eta_0 \log \widehat{\boldsymbol{k}}_{\widehat{\sigma}(i)} + \mathbf{1}_{\{\mathrm{k_i}=1\}} \mathcal{L}_{box}(\boldsymbol{b}_i, \widehat{\boldsymbol{b}}_{\widehat{\sigma}(i)}) \Big], \tag{18}$$

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 Rezatofighi et al. (2019) 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 18 are set to 1, 2, 5 following Carion et al. (2020), respectively.

B More Ablations

Hyper-parameters. We follow Carion et al. (2020) to set loss weight of $\mathcal{L}_{loc}$ (λ_2, see Equation 13) to 1.0 in all setups and further ablate the λ_0 and λ_1 using Conditional DETR on COCO. Figure 7 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.

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 5, Siamese DETR outperforms UP-DETR by a significant margin of 2.9/5.1/6.7 AR with detecting top 1/10/100 object(s), respectively. It suggests Siamese DETR provides a better initialization in downstream tasks than UP-DETR due to its localization and discrimination prior.

Table 6: Memory cost and iteration time. *Notice that Siamese DETR and UP-DETR are implemented using MMSelfSup, and DETReg are implemented using released repository. Several hooks in MMSelfSup may slow down the iteration time, therefore the iteration time of DETReg is for reference only.

Method	GPU Mem.	Iteration time
Siamese DETR	8428 MB	0.8018 s/it
UP-DETR	8199 MB	0.7524 s/it
DETReg*	10528 MB	0.5028 s/it

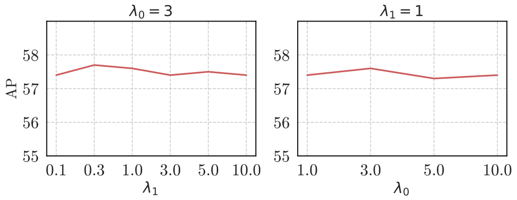

Figure 7: Sensitivity of Hyper-parameters in Siamese DETR.

Table 5: Ablations on downstream initialization. We initialize the Vanilla DETR using models pretrained by Siamese DETR 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
ours	12.4	23.0	30.7

Data Efficiency. 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 splits are selected randomly. As shown in Figure 8, 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.

Memory Cost and Iteration Time. The parameters in Siamese DETR are all shared, therefore it only has a slight increase on GPU memory. Besides, performing multi-view learning and adding crop-level features does not bring too much time cost. We provided a quantitive results in Table 6.

C More Visualization

C.1 Convergence

Figure 9 and Figure 10 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 11. The visualization results verify better transferability of Siamese DETR against its counterpart.

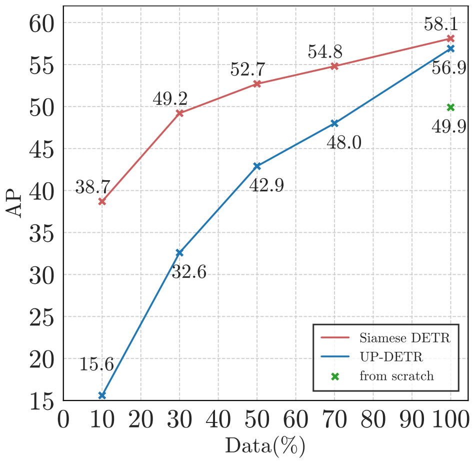

Figure 8: Data efficiency of Siamese DETR. We finetune Siamese DETR and UP-DETR using 10%/30%/50%/70% PASCAL VOC datasets.

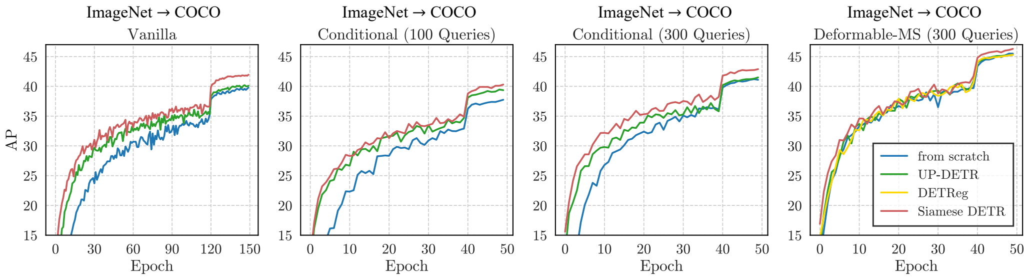

Figure 9: Illustration of convergence curves when finetuned on COCO.

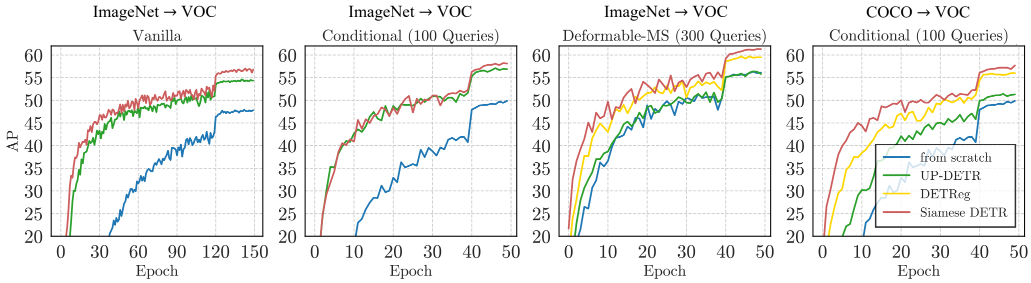

Figure 10: Illustration of convergence curves when finetuned on PASCAL VOC.

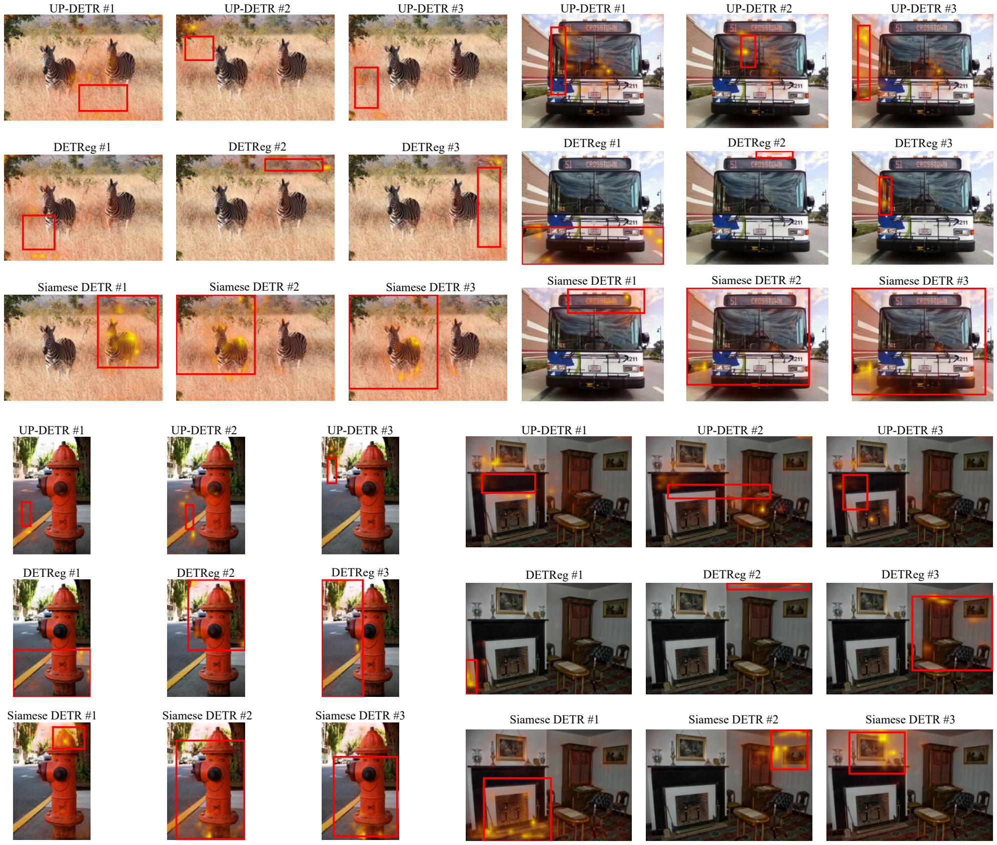

Figure 11: More visualization on box predictions and attention maps when initializing the downstream models using Siamese DETR, UP-DETR and DETReg without finetuning.