Revisiting DETR Pre-training for Object Detection

Yan Ma[1] · Weicong Liang[2] · Bohan Chen[3] · Yiduo Hao[6] ·
Bojian Hou[4] Xiangyu Yue[5] · Chao Zhang[2] · Yuhui Yuan[6]

Abstract Motivated by the remarkable achievements of DETR-based approaches on COCO object detection and segmentation benchmarks, recent endeavors have been directed towards elevating their performance through self-supervised pre-training of Transformers while preserving a frozen backbone. Noteworthy advancements in accuracy have been documented in certain studies. Our investigation delved deeply into a representative approach, DETReg, and its performance assessment in the context of emerging models like $\mathcal{H}$-Deformable-DETR. Regrettably, DETReg proves inadequate in enhancing the performance of robust DETR-based models under full data conditions. To dissect the underlying causes, we conduct extensive experiments on COCO and PASCAL VOC probing elements such as the selection of pre-training datasets and strategies for pre-training target generation. By contrast, we employ an optimized approach named Simple Self-training which leads to marked enhancements through the combination of an improved box predictor and the Objects365 benchmark. The culmination of these endeavors results in a remarkable AP score of 59.3% on the COCO `val` set, outperforming $\mathcal{H}$-Deformable-DETR + Swin-L without pre-training by 1.4%. Moreover, a series of synthetic pre-training datasets, generated by merging contemporary image-to-text(LLaVA) and text-to-image (SDXL) models, significantly amplifies object detection capabilities.

Keywords Object detection, DETR, Pre-training

1 Introduction

Recently, the DETR-based approaches (Carion et al., 2020; Jia et al., 2023; Li et al., 2023; Zhang et al., 2022; Zhu et al., 2020) have achieved significant progress and pushed the frontier on both object detection and segmentation tasks. For example, DINO-DETR (Zhang et al., 2022), $\mathcal{H}$-Deformable-DETR (Jia et al., 2023), and Group-DETRv2 (Chen et al., 2022) have set new state-of-the-art object detection performance on COCO benchmark. Mask-DINO (Li et al., 2023) further extends DINO-DETR and establishes the best results across COCO instance segmentation and panoptic segmentation tasks. To some degree, this is the first time that end-to-end transformer approaches can achieve an even better performance than the conventional heavily tuned strong detectors (Li et al., 2021; Liu et al., 2022b) based on convolution, e.g., Cascade Mask-RCNN and HTC++.

Despite the great success of these DETR-based approaches, they still choose a randomly initialized Transformer and thus fail to unleash the potential of a fully pre-trained detection architecture like (Wei et al., 2021), which already verifies the benefits of aligning the pre-training architecture with the downstream architecture. Figure 1a and 1b illustrate the distribution of the number of parameters and GFLOPs within a standard Deformable-DETR network based on ResNet50 backbone. We can see that the Transformer encoder and decoder occupy 34% of the parameters and 65% of the GFLOPs, which means there exists much room for improvement along the path of performing pre-training on the Transformer part within DETR.

Several recent works have improved DETR-based object detection models by performing self-supervised pre-training on the Transformer encoder and decoder while freezing the backbone. For example, UP-DETR (Dai et al., 2021) pre-trains Transformer to detect random patches in an image, DETReg (Bar et al., 2022) pre-trains Transformer to match object locations and features with priors generated from Selective Search algorithm, and most recently, Siamese DETR locates the target boxes with the query features extracted from a different view's corresponding box. However, these works utilize either the vanilla DETR network (AP=42.1% in terms of object detection performance

[1]University of Toronto   [2]Peking University   [3]Xi'an Jiaotong-Liverpool University   [4]University of Pennsylvania   [5]CUHK   [6]Microsoft Research Asia   ✉ yuhui.yuan@microsoft.com

(a) #parameters

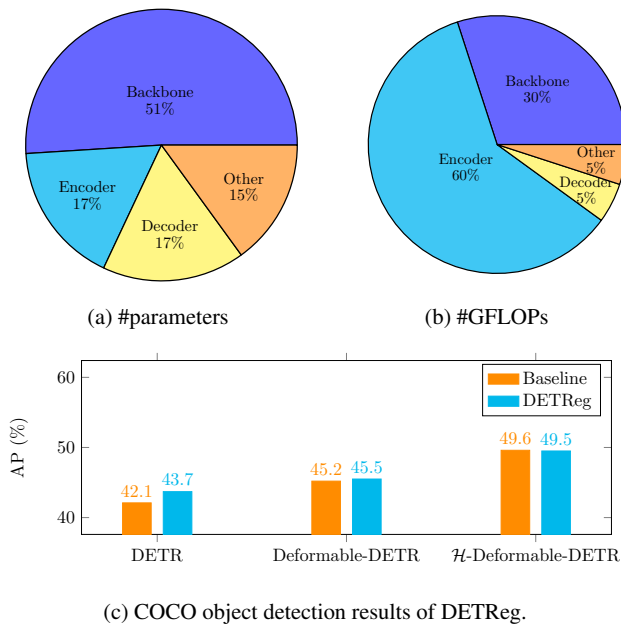(b) #GFLOPs

(c) COCO object detection results of DETReg.

Fig. 1: **The distribution of the number of parameters and GFLOPs within Deformable-DETR network with a ResNet**50 **backbone, and the pre-training performance of DETReg.** As shown in (a) and (b), we can see that around 34% parameters and 65% GFLOPs are distributed in the randomly initialized Transformer encoder and decoder. According to (c), DETReg only improves the vanilla DETR and Deformable-DETR by +1.6% and +0.3% while showing no gains over the stronger $\mathcal{H}$-Deformable-DETR.

on COCO) or the Deformable-DETR variant (AP=45.2%). Their results fall significantly short when pre-training on the latest much stronger DETR model like $\mathcal{H}$-Deformable-DETR (Jia et al., 2023) (AP=49.6%). In Figure 1c, we present the object detection results of different DETR models on COCO under two conditions: without pre-training of the Transformer component (referred to as the *baseline*) and with pre-training using the DETReg method. In both cases, the backbones of these models are ResNet50 initialized with SwAV (Caron et al., 2020). Notably, in the case of the $\mathcal{H}$-Deformable-DETR, the utilization of the DETReg pre-training actually leads to a performance decrease rather than an improvement.

In this work, we first take a closer look at how much self-supervised pre-training methods, exemplified by DETReg, can improve over the increasingly potent DETR models on COCO object detection benchmark. Our investigation unveils a significant limitation in the efficacy of DETReg when applied to fortified DETR networks bolstered by improvements like the SwAV pre-trained backbone, deformable techniques in Deformable-DETR, and the hybrid matching scheme in $\mathcal{H}$-Deformable-DETR. We pinpoint the crux of the issue as originating from unreliable box proposals generated by unsupervised methods like Selective Search, which contribute to noisy localization targets, and

the weak semantic information provided through feature reconstruction which is not an efficient classification target either. These drawbacks make the self-supervised pre-training methods ineffective when applied to an already strong DETR model.

To fix this, we propose to use a COCO object detector to get more accurate pseudo-boxes with informative pseudo-class labels. Extensive ablation experiments underscore the impact of three pivotal factors: the choice of pre-training datasets (ImageNet vs. Objects365), localization pre-training targets (Selective Search proposals vs. pseudo-box predictions), and classification pre-training targets (object-embedding vs. pseudo-class predictions).

Our findings reveal that a Simple Self-training scheme, employing pseudo-box and pseudo-class predictions as pre-training targets, outperforms the DETReg approach in various settings. Notably, this simple design yields discernible pre-training enhancements even for the state-of-the-art DETR network without accessing the pre-training benchmark's ground-truth label. For example, with a ResNet50 backbone and the Objects365 pre-training dataset, Simple Self-training elevates DETReg's COCO object detection results on $\mathcal{H}$-Deformable-DETR by 3.6%. Furthermore, a remarkable performance is observed with the Swin-L backbone, yielding competitive results 59.3%.

Additionally, we delve into an exploration of contemporary image-to-text and text-to-image generation models, aiming to create a sequence of synthetic datasets for object detection pre-training. Empirically, our observations yield encouraging outcomes, as pre-training with these synthetic datasets demonstrates commendable performance even when compared against the widely adopted Objects365 benchmark, which entails substantial annotation costs. In general, our efforts are poised to provide a more authentic assessment of the progress in the formidable task of DETR pre-training.

2 Related Work

DETR for object detection. Since the emergence of DETR (Carion et al., 2020) as the first fully end-to-end object detector, many works have extended DETR with novel techniques to achieve state-of-the-art results on various vision tasks. To accelerate the convergence of the original DETR, Deformable-DETR (Zhu et al., 2020) proposes a novel multi-scale deformable self/cross-attention to focus on a sparse set of important sampling points around a reference point. Furthermore, based on DAB-DETR (Liu et al., 2022a) with a different query formulation, DINO-DETR (Zhang et al., 2022) introduces a query denoising scheme and sets new records on object detection tasks. Besides, to address the training efficiency bottleneck caused by one-to-one matching in DETR, $\mathcal{H}$-Deformable-DETR (Jia

et al., 2023) and Group-DETR (Chen et al., 2022) propose to train with more queries in the transformer decoder with an additional one-to-many matching scheme, which helps to achieve even faster convergence and better performance.

Self-supervised pre-training. Self-supervised learning (SSL) has achieved remarkable results in image classification methods such as MoCo (He et al., 2020), SimCLR (Chen et al., 2020), and BYOL (Grill et al., 2020). However, SSL on object detection has shown limited transferability. To overcome this challenge, many works have proposed pretext tasks that leverage region or pixel localization cues to enhance the pre-training signals. For example, InsLoc (Yang et al., 2021a) uses contrastive learning on foreground patches to learn instance localization. UniVIP (Li et al., 2022) exploits scene similarity, scene-instance correlation, and instance discrimination to capture semantic affinity. CP^2 (Wang et al., 2022) employs pixel-wise contrastive learning to facilitate both image-level and pixel-level representation learning. Unlike most of these methods that aim to improve conventional object detectors such as Faster R-CNN or Cascade R-CNN, we focus on designing an effective pre-training scheme for the state-of-the-art DETR-based detector.

DETR pre-training. DETR typically relies on a supervised pre-trained backbone on ImageNet and random initialization of the transformer encoder and decoder. Some recent works have explored pre-training the transformer component of DETR for enhanced object detection performance. For example, UP-DETR (Dai et al., 2021) introduces an unsupervised pretext task to detect and reconstruct random patches of the input. DETReg (Bar et al., 2022) refines the pretext task by using unsupervised region proposals from Selective Search (Uijlings et al., 2013) instead of random patches and also reconstructs the object embeddings of these regions from its SwAV (Caron et al., 2020) backbone to learn invariant representations. Siamese DETR (Huang et al., 2023) employs a siamese self-supervised learning approach to pre-train DETR in a symmetric pipeline where each branch takes one view as input and aims to locate and discriminate the corresponding regions from another view. However, these pre-training methods only yield minor improvements to a strong DETR variant like Deformable-DETR.

Self-training. Self-training is a powerful technique for improving various computer vision tasks, such as image classification (Li et al., 2023; Sahito et al., 2022), object detection (Vandeghen et al., 2022; Yang et al., 2021b), and segmentation (Zhu et al., 2021). A common self-training method is NoisyStudent (Xie et al., 2020), which trains a teacher model on labeled data and uses it to generate pseudo-labels on unlabeled images. These pseudo-labels are then used to train a student model, and this process is repeated to obtain better models by updating the teacher model with the previous student model. The ASTOD (Vandeghen et al., 2022) framework applies an iterative self-training process for object detection, using multiple image views to produce high-quality pseudo-labels. ST++(Yang et al., 2022) is a recent self-training algorithm for segmentation tasks, which uses confidence scores to filter out incorrect pseudo-labels. (Zoph et al., 2020) has demonstrated that self-training outperforms traditional pre-training methods in various scenarios, including low and high data regimes, and can even succeed when pre-training methods fail. Unlike these complex self-training schemes that use an iterative approach to refine pseudo-labels, we propose a Simple Self-training scheme that generates pseudo-labels only once by keeping a fixed number of the most confident predictions.

3 Approach

In this work, we focus on studying how to perform pre-training over the Transformer encoder and decoder parts within DETR for object detection tasks following (Bar et al., 2022; Dai et al., 2021). The goal of DETR pre-training is to design an effective pretext task that can make the best use of a large-scale unlabeled dataset that has no ground-truth bounding box annotations.

3.1 Formulation

The conventional DETR model has three components, the backbone extracting the image feature, the encoder enhancing the feature with a self-attention mechanism, and the decoder turning query inputs into object class and location predictions through cross-attention with image features. The existing self-supervised pre-training methods share a similar scheme that optimizes the encoder and decoder network parameters on the pre-training dataset while freezing a pre-trained backbone. After pre-training, all three components are tuned together on the downstream dataset. The pipeline is illustrated in Figure 2.

Preliminary. In the following article, we formulate the general self-supervised pre-training process as several equations. We use f_{θ_B}, f_{θ_E}, f_{θ_D} to represent the backbone, Transformer encoder, and Transformer decoder within a DETR network parameterized by θ_B, θ_E, and θ_D. The input images from the pre-training and downstream dataset are denoted as $\overline{\mathbb{X}} = \{\overline{\mathbf{x}}_1, \cdots, \overline{\mathbf{x}}_N\}$ and $\mathbb{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ respectively, where $N \gg M$. The ground-truth label of downstream data is $\mathbb{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_M | \mathbf{y}_i = (\mathbf{c}_i, \mathbf{b}_i)\}$, where $\mathbf{c}_i$ is the category label and $\mathbf{b}_i$ is the box location label. Typically, the domain-specific pre-training data labels are lack-
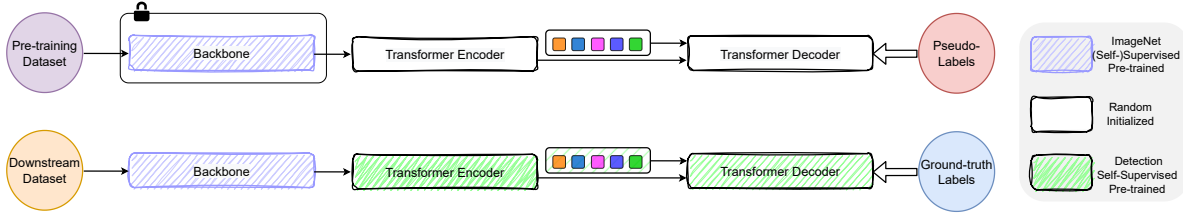
Fig. 2: **The overall framework of self-supervised pre-training scheme**. There are two steps to pre-train the DETR network. In the first step, we freeze the backbone and pre-train a randomly initialized Transformer encoder and decoder with the well-designed pre-training target on a large-scale pre-training benchmark. In the second step, we initialize the encoder and decoder with pre-trained weights and fine-tune all the parameters of the DETR network on the downstream dataset supervised by ground-truth labels.

ing and most works choose to generate the pseudo-labels, i.e., $\overline{\mathbb{Y}} = \{\overline{\mathbf{y}}_1, \cdots, \overline{\mathbf{y}}_N\}$ instead.

Pre-train. We illustrate the mathematical formulations of the DETR pre-training with Equation 1 and 2. Specifically, the pre-training input $\overline{\mathbf{x}}_i$ is forwarded through the backbone f_{θ_B}, encoder f_{θ_E}, and decoder f_{θ_D} to get the prediction $\overline{\mathbf{z}}_i$. Here $\theta_\mathsf{B}, \theta_\mathsf{E}, \theta_\mathsf{D}$ represent the learnable parameters for the three network components respectively. θ_B is initialized with SwAV (Caron et al., 2020) self-supervised pre-training method and frozen during pre-training. θ_E and θ_D are randomly initialized and then optimized to minimize the pre-training loss $\mathcal{L}_{\text{pre}}(\cdot)$, which is calculated with network output $\overline{\mathbf{z}}_i$ and pre-training target $\overline{\mathbf{y}}_i$.

$$\overline{\mathbf{z}}_i = f_{\theta_\mathsf{D}}(f_{\theta_\mathsf{E}}(f_{\theta_\mathsf{B}}(\overline{\mathbf{x}}_i)), \mathbb{Q}), \tag{1}$$

$$\widehat{\theta}_\mathsf{D}, \widehat{\theta}_\mathsf{E}, \widehat{\mathbb{Q}} = \operatorname*{argmin}_{\theta_\mathsf{D}, \theta_\mathsf{E}, \mathbb{Q}} \sum_{i=1}^N \mathcal{L}_{\text{pre}}(\overline{\mathbf{z}}_i, \overline{\mathbf{y}}_i), \tag{2}$$

where $\mathbb{Q} = \{\mathbf{q}_1, \cdots, \mathbf{q}_k\}$ represents the learnable object query of decoder and will also be jointly optimized with the encoder/decoder parameters. $\widehat{\theta}_\mathsf{D}, \widehat{\theta}_\mathsf{E}, \widehat{\mathbb{Q}}$ represent the decoder parameters, encoder parameters, and object query after pre-training. In the following section 3.2, we will illustrate the formulation of $\mathcal{L}_{\text{pre}}$ in different methods.

Fine-tune. We obtain the optimized encoder and decoder parameter $\widehat{\theta}_\mathsf{E}, \widehat{\theta}_\mathsf{D}$ during pre-training. Then we tune the same network on the downstream data $\mathbf{x}_i$. Here, we initialize the backbone, encoder and decoder parameter with $\theta_\mathsf{B}, \widehat{\theta}_\mathsf{E}, \widehat{\theta}_\mathsf{D}$, and denote the network output as $\mathbf{z}_i$. All parameters of the three components and learnable query $\mathbb{Q}$ are optimized to minimize the downstream loss $\mathcal{L}_{\text{ds}}(\cdot)$ between $\mathbf{z}_i$ and downstream label $\mathbf{y}_i$.

$$\mathbf{z}_i = f_{\widehat{\theta}_\mathsf{D}}(f_{\widehat{\theta}_\mathsf{E}}(f_{\theta_\mathsf{B}}(\mathbf{x}_i)), \widehat{\mathbb{Q}}), \tag{3}$$

$$\widetilde{\theta}_\mathsf{D}, \widetilde{\theta}_\mathsf{E}, \widetilde{\theta}_\mathsf{B}, \widetilde{\mathbb{Q}} = \operatorname*{argmin}_{\widehat{\theta}_\mathsf{D}, \widehat{\theta}_\mathsf{E}, \theta_\mathsf{B}, \widehat{\mathbb{Q}}} \sum_{i=1}^M \mathcal{L}_{\text{ds}}(\mathbf{z}_i, \mathbf{y}_i), \tag{4}$$

where $\widetilde{\theta}_\mathsf{D}, \widetilde{\theta}_\mathsf{E}, \widetilde{\theta}_\mathsf{B}, \widetilde{\mathbb{Q}}$ are optimized decoder, encoder, backbone parameters, and object query after downstream fine-tuning.

3.2 Instantiations

Assume the target of the i-th pre-training input can be denoted as $\overline{\mathbf{y}}_i = \{\overline{\mathbf{y}}_{i1}, \cdots, \overline{\mathbf{y}}_{im}\}$, where m is the number of objects in each target. The network output consists of k bounding box predictions, which is the same as the number of object queries. We denote the corresponding prediction as $\overline{\mathbf{z}}_i = \{\overline{\mathbf{z}}_{i1}, \cdots, \overline{\mathbf{z}}_{ik}\}$. Typically, the number of targets in $\overline{\mathbf{y}}_i$ is less than 30, while we set our DETR network to output 100 or 300 predictions, so $m < k$. Thus we pad the targets with no-object category $\varnothing$ following DETR (Carion et al., 2020) to be of size k. Then, DETR performs one-to-one alignment via Hungarian bipartite matching algorithm (Kuhn, 1955) over $\overline{\mathbf{y}}_i$ and $\overline{\mathbf{z}}_i$. We illustrate the mathematical formulation in Equation 5, which computes the optimal label assignment for each prediction by minimizing the matching cost function $\mathcal{L}_{\text{match}}(\cdot)$:

$$\sigma_i = \operatorname*{argmin}_{\sigma_i \in \Sigma_k} \sum_{j=1}^k \mathcal{L}_{\text{match}}(\overline{\mathbf{y}}_{ij}, \overline{\mathbf{z}}_{i\sigma_i(j)}), \tag{5}$$

where Σ_k represents all permutations over k elements and $\sigma_i(j)$ maps the targeted box j to the most similar predicted box within the i-th input. The matching cost function $\mathcal{L}_{\text{match}}(\cdot)$ measures the predictions from two aspects including the localization accuracy and classification accuracy following DETR (Carion et al., 2020).

Most self-supervised pre-training methods differentiate through the design of pretext tasks, which results in different structures for the pre-training target $\overline{\mathbf{y}}_i$ and implementations of the pre-training loss $\mathcal{L}_{\text{pre}}$. A good pretext task design can improve the final prediction performance. In the following, we first introduce the instantiation of a representative method called DETReg (Bar et al., 2022). Then, we propose two more effective pre-training schemes: DETReg + Pseudo-box and Simple Self-training. Both methods focus on enhancing the localization and classification pre-training target quality. We compare the pre-training pipeline of three methods in Figure 3.

DETReg. DETReg uses an unsupervised region proposal method named *Selective Search (ss)* to generate the tar-

get boxes. The j-th "box proposal" for the i-th input is denoted as $\overline{\mathbf{b}}_{ij}^{ss} \in [0,1]^4$. We select the top $\overline{k}$ Selective Search box proposals $\{\overline{\mathbf{b}}_{i1}^{ss}, \cdots, \overline{\mathbf{b}}_{i\overline{k}}^{ss}\}$ and pair them with the binary category target padded to the size of network query number k ($k > \overline{k}$) $\{\overline{\mathbf{p}}_{i1}^{ss}, \cdots, \overline{\mathbf{p}}_{ik}^{ss} | \overline{\mathbf{p}}_{i1}^{ss}, \cdots, \overline{\mathbf{p}}_{i\overline{k}}^{ss} = 1, \overline{\mathbf{p}}_{i(\overline{k}+1)}^{ss}, \cdots, \overline{\mathbf{p}}_{ik}^{ss} = 0\}$, where $\overline{\mathbf{p}}_{ij}^{ss} = 1$ indicates the element is a box proposal while $\overline{\mathbf{p}}_{ij}^{ss} = 0$ indicates a padded $\varnothing$. To compensate for the lack of semantic information in the binary category, the DETReg network incorporates another object embedding reconstruction branch to predict the object embeddings $\{\overline{\mathbf{f}}_{i1}, \cdots, \overline{\mathbf{f}}_{ik} | \overline{\mathbf{f}}_{ij} \in \mathbb{R}^d\}$ of detected boxes, which is supervised by the target object descriptor $\{\overline{\mathbf{f}}_{i1}^{\text{swav}}, \cdots, \overline{\mathbf{f}}_{i\overline{k}}^{\text{swav}}\}$ with $\overline{\mathbf{f}}_{ij}^{\text{swav}}$ indicating the object embedding extracted from the image patch in the j-th box proposal on the i-th input with a fixed SwAV backbone. Therefore, the pre-training target and network prediction are denoted as Equation 6:

$$\overline{\mathbf{y}}_{ij} = (\overline{\mathbf{p}}_{ij}^{ss}, \overline{\mathbf{b}}_{ij}^{ss}, \overline{\mathbf{f}}_{ij}^{\text{swav}}), \quad \overline{\mathbf{z}}_{ij} = (\overline{\mathbf{p}}_{ij}, \overline{\mathbf{b}}_{ij}, \overline{\mathbf{f}}_{ij}). \tag{6}$$

The pre-training loss is the sum of binary classification loss $\mathcal{L}_{\text{cls}}^{\text{bin}}(\cdot)$, box loss $\mathcal{L}_{\text{box}}(\cdot)$, and embedding loss $\mathcal{L}_{\text{emb}}(\cdot)$ through all k outputs as below:

$$\begin{aligned} \mathcal{L}_{\text{pre}}(\overline{\mathbf{y}}_i, \overline{\mathbf{z}}_i) = &\sum_{j=1}^{k} \lambda_c \mathcal{L}_{\text{cls}}^{\text{bin}}(\overline{\mathbf{p}}_{ij}^{ss}, \overline{\mathbf{p}}_{i\sigma_i(j)}) \\ &+ \lambda_b \mathbb{1}_{\{\overline{\mathbf{p}}_{ij}^{ss} \neq 0\}} \mathcal{L}_{\text{box}}(\overline{\mathbf{b}}_{ij}^{ss}, \overline{\mathbf{b}}_{i\sigma_i(j)}) \\ &+ \lambda_e \mathcal{L}_{\text{emb}}(\overline{\mathbf{f}}_{ij}^{\text{swav}}, \overline{\mathbf{f}}_{i\sigma_i(j)}), \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{cls}}^{\text{bin}}(\cdot)$ is the binary classification loss which can be implemented as Cross Entropy Loss or Focal Loss. $\mathcal{L}_{\text{box}}(\cdot)$ is the sum of L1 and GIoU Loss, and $\mathcal{L}_{\text{emb}}(\cdot)$ is the L1 Loss. λ_c, λ_b, and λ_e are loss coefficients and $\sigma_i(j)$ maps the target box j to the assigned predicted box $\sigma_i(j)$ with lowest cost within the i-th input.

DETReg + Pseudo-box. The unsupervised box proposals like Selective Search boxes are of very low quality. To handle this, we employ two off-the-shelf well-trained COCO object detectors to predict the pseudo-boxes for the pre-training data to replace the Selective Search proposals. Specifically, we replace the $(\overline{\mathbf{p}}_{ij}^{ss}, \overline{\mathbf{b}}_{ij}^{ss})$ in Equation 6 and 7 with $(\overline{\mathbf{p}}_{ij}^{\text{pseudo}}, \overline{\mathbf{b}}_{ij}^{\text{pseudo}})$. We use $\mathcal{H}$-Deformable-DETR with ResNet50 or Swin-L backbone as our detector network. We first train them on COCO, then use the trained detector to predict pseudo-boxes on the pre-training dataset, and the top 30 predictions are selected as $\overline{k}$.

Simple Self-training. We further replace the binary category target $\overline{\mathbf{p}}_{ij}^{\text{pseudo}}$ with category predictions $\overline{\mathbf{c}}_{ij}^{\text{pseudo}} \in \{\varnothing, c_1, \cdots, c_n\}$ of aforementioned COCO object detectors as the classification target and remove $\overline{\mathbf{f}}_{ij}^{\text{swav}}$ since we already have detailed class information. Due to that the detector is

Localization method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR@10	AR@30	AR@100
Selective Search	0.5	1.6	0.2	0.2	0.3	1.2	3.7	8.3	15.5
$\mathcal{H}$-Deformable-DETR + R50	28.4	40.4	30.2	12.7	26.7	43.1	26.5	37.4	**47.7**
$\mathcal{H}$-Deformable-DETR + Swin-L	**30.7**	**41.3**	**33.0**	**15.2**	**29.0**	**44.9**	**28.1**	**38.5**	47.4

Table 1: Objects356 AP and AR score for Selective Search box proposals, and pseudo-box predictions of $\mathcal{H}$-Deformable-DETR-based COCO detectors with R50 and Swin-L backbone.

trained on COCO and the pseudo-category labels it predicts are the 80 COCO categories, the binary classification turns into a multiclass classification. The formulation is shown below:

$$\overline{\mathbf{y}}_{ij} = (\overline{\mathbf{c}}_{ij}^{\text{pseudo}}, \overline{\mathbf{b}}_{ij}^{\text{pseudo}}), \quad \overline{\mathbf{z}}_{ij} = (\overline{\mathbf{c}}_{ij}, \overline{\mathbf{b}}_{ij}), \tag{8}$$

$$\begin{aligned} \mathcal{L}_{\text{pre}}(\overline{\mathbf{y}}_i, \overline{\mathbf{z}}_i) = &\sum_{j=1}^{k} \lambda_c \mathcal{L}_{\text{cls}}^{\text{mul}}(\overline{\mathbf{c}}_{ij}^{\text{pseudo}}, \overline{\mathbf{c}}_{i\sigma_i(j)}) \\ &+ \lambda_b \mathbb{1}_{\{\overline{\mathbf{c}}_{ij}^{\text{pseudo}} \neq \varnothing\}} \mathcal{L}_{\text{box}}(\overline{\mathbf{b}}_{ij}^{\text{pseudo}}, \overline{\mathbf{b}}_{i\sigma_i(j)}), \end{aligned} \tag{9}$$

where $\mathcal{L}_{\text{cls}}^{\text{mul}}(\cdot)$ is the multiclass classification loss.

3.3 Discussion

We utilize ImageNet and Objects365 as the two pre-training benchmarks. To display the quality of Selective Search proposals and pseudo-boxes generated by two off-the-shelf COCO object detectors, we report their boxes' Average Precision and Average Recall on Objects365 validation set in Table 1. As can be seen, pseudo-boxes generated by COCO object detectors are far more accurate than Selective Search boxes. We also visualize their box proposals in Figure 4.

Unlike the conventional self-training scheme (Xie et al., 2020; Zoph et al., 2020) that relies on applying complicated augmentation strategy to boost the quality of pseudo-labels, adjusting NMS threshold carefully, and re-generating more accurate pseudo-labels based on the fine-tuned models in an iterative manner, our Simple Self-training method directly generate the pseudo-labels for one time without those tricks, resulting in a much simpler approach.

4 Experiment

4.1 Implementation Details

Datasets. Our object detection network is pre-trained on the ImageNet or Objects365 (Shao et al., 2019) benchmark, then fine-tuned on COCO train2017 and evaluated on COCO val2017, or fine-tuned on PASCAL VOC trainval07+12 and evaluated on PASCAL VOC test2007. For the pre-training benchmarks, ImageNet has 1.2 Million images
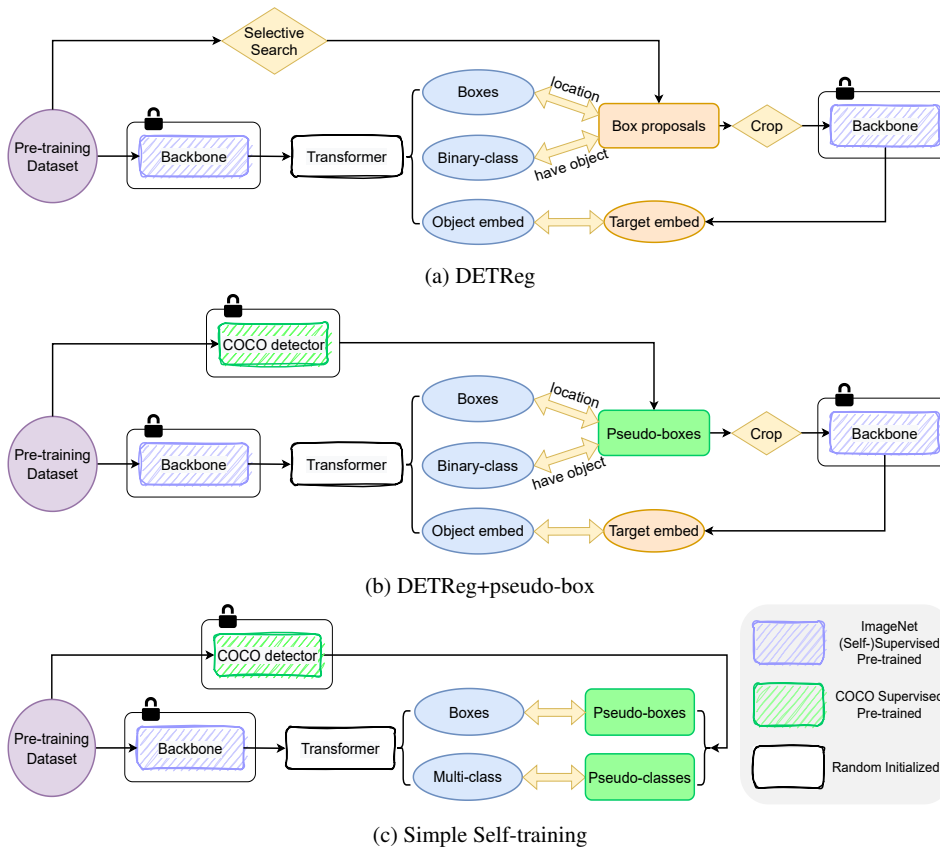
(a) DETReg

(b) DETReg+pseudo-box

(c) Simple Self-training

Fig. 3: **The pre-training pipelines of DETReg, DETReg+pseudo-box, and Simple Self-training.** In DETReg and DETReg+pseudo-box, we use an extra frozen backbone branch to get the target object embeddings from the image crops. The binary-class outputs of the Transformer predict whether the detected boxes contain an object.
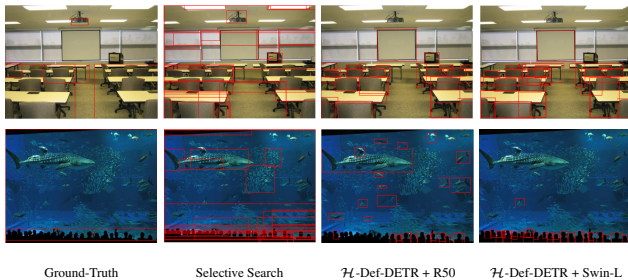

Fig. 4: **Qualitative comparisons of the top** 30 **generated bounding boxes of different methods on Objects**365**.** The methods include Selective Search and trained $\mathcal{H}$-Deformable-DETR detectors with R50 or Swin-L backbones.

which mostly contain one object since the dataset is created for classification. Objects365 is a large-scale dataset for object detection with 2 Million images. The image scene is more complicated with around 15 ground-truth bounding boxes per image on average. We use Objects365 as the default pre-training benchmark for all experiments in sections 4.2 and 4.4, as its complex scenes bring better pre-training performance for the Simple Self-training approach.

Architectures. We use two kinds of DETR backbones including ResNet50 which is self-supervised pre-trained by SwAV on ImageNet and Swin-L which is supervised pre-trained on ImageNet. We pre-train three DETR-based architectures in Section 4.3 including vanilla DETR (Carion et al., 2020), Deformable-DETR (Zhu et al., 2020), and $\mathcal{H}$-Deformable-DETR (Jia et al., 2023), which is a recent state-of-the-art object detector based on a combination of an improved Deformable-DETR and an effective hybrid matching scheme. The Transformer module in those architectures is composed of 6 encoder layers and 6 decoder layers. The vanilla DETR and Deformable-DETR are plain without tricks, while $\mathcal{H}$-Deformable-DETR is improved with iterative bounding box refinement, two-stage (Zhu et al., 2020), mixed query selection, and look forward twice scheme (Zhang et al., 2022). By default, we use $\mathcal{H}$-Deformable-DETR with ResNet50 backbone for the ablation study.

Training. We pre-train the network on ImageNet for 5 epochs following DETReg or on Objects365 for 3 epochs to ensure the same iteration number according to their different dataset sizes. For fine-tuning, we train for 150 epochs with

Method	Framework	Backbone	#epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Swin (Liu et al., 2021)	HTC	Swin-L	36	57.1	75.6	62.5	42.4	60.7	71.1
Group-DETR (Chen et al., 2022)	DETR	Swin-L	36	58.4	-	-	41.0	62.5	73.9
DINO-DETR (Zhang et al., 2022)	DETR	Swin-L	36	58.5	77.0	64.1	41.5	62.3	**74.0**
$\mathcal{H}$-Deformable-DETR (Jia et al., 2023)	DETR	Swin-L	36	57.9	76.9	63.7	42.4	61.9	73.4
Ours (pre-trained $\mathcal{H}$-Deformable-DETR)	DETR	Swin-L	24	**59.3**	**77.9**	**65.1**	**44.1**	**62.9**	73.6

Table 2: System-level comparisons with the state-of-the-art DETR-based single-scale evaluation results on COCO val set.

Method	DETR model	Pretrain	#query	#epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
from scratch	DETR	-	100	150	40.3	61.3	42.2	18.2	44.6	60.5
DETRReg	DETR	ImageNet	100	150	40.2	60.7	42.3	17.6	44.3	59.6
ours	DETR	ImageNet	100	150	**41.9**	**62.7**	**44.0**	**20.7**	**46.0**	**62.8**
from scratch	DDETR-MS	-	300	50	45.2	64.2	49.4	**27.2**	49.3	59.1
DETRReg	DDETR-MS	ImageNet	300	50	43.5	61.4	47.3	24.2	47.1	58.7
ours	DDETR-MS	ImageNet	300	50	**46.0**	**64.4**	**50.0**	26.6	**49.8**	**61.5**
from scratch	$\mathcal{H}$-DDETR-MS	-	300	12	49.6	67.5	54.1	31.9	53.3	64.1
DETRReg	$\mathcal{H}$-DDETR-MS	ImageNet	300	12	49.5	66.8	53.9	30.5	53.5	63.6
ours	$\mathcal{H}$-DDETR-MS	ImageNet	300	12	**51.6**	**69.4**	**56.4**	**35.0**	**55.3**	**66.8**

Table 3: Comparisons with self-supervised pre-training method DE-TReg on the COCO downstream benchmark.

Method	DETR model	Pretrain	#query	#epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
from scratch	DETR	-	100	150	56.3	80.3	60.6	10.2	36.0	65.9
DETRReg	DETR	ImageNet	100	150	60.9	82.0	65.9	15.1	40.8	69.8
ours	DETR	ImageNet	100	150	**63.5**	**83.8**	**68.6**	**22.5**	**44.3**	**72.1**
from scratch	DDETR-MS	-	300	50	61.1	83.1	68.0	25.5	47.4	67.7
DETRReg	DDETR-MS	ImageNet	300	50	63.6	82.6	70.2	27.5	49.7	70.2
ours	DDETR-MS	ImageNet	300	50	**67.8**	**85.4**	**75.5**	**30.9**	**54.7**	**74.4**
from scratch	$\mathcal{H}$-DDETR-MS	-	300	12	63.8	82.4	70.0	26.5	50.0	70.4
DETRReg	$\mathcal{H}$-DDETR-MS	ImageNet	300	12	67.7	84.5	74.9	**35.1**	55.1	74.7
ours	$\mathcal{H}$-DDETR-MS	ImageNet	300	12	**71.6**	**87.0**	**79.2**	33.1	**60.3**	**78.2**

Table 4: Comparisons with self-supervised pre-training method DE-TReg on the PASCAL VOC downstream benchmark.

Method	Pre-training dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETRReg	ImageNet	**49.5**	**66.8**	**53.9**	30.5	**53.5**	**63.6**
	O365	49.2	66.5	53.6	**31.4**	53.2	63.5
DETRReg+pseudo-box	ImageNet	50.9	68.3	55.7	33.6	54.6	64.9
	O365	**52.0**	**69.6**	**56.7**	**36.1**	**55.9**	**65.3**
Simple Self-training	ImageNet	51.6	69.4	56.4	35.0	55.3	66.8
	O365	**52.8**	**70.9**	**57.6**	**37.0**	**56.6**	**67.3**

Table 5: Effect of pre-training dataset choices.

Method	Localization target	Classification target	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
from scratch	-	-	49.6	67.5	54.1	31.9	53.3	64.1
DETRReg	Selective Search	Object-embedding loss	49.2	66.5	53.6	31.4	53.2	63.5
DETRReg+pseudo-box	Pseudo-box prediction	Object-embedding loss	52.0	69.6	56.7	36.1	55.9	65.3
Simple Self-training	Pseudo-box prediction	Pseudo-class prediction	52.8	70.9	57.6	37.0	56.6	67.3
Supervised	Ground-truth	Ground-truth	**53.2**	**71.5**	**58.1**	**37.3**	**57.0**	**67.4**

Table 6: Fine-tuning results on COCO after pre-training with different methods using various localization and classification pre-training targets on Objects365.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
from scratch	63.8	82.4	70.0	26.5	50.0	70.4
DETRReg	67.7	84.7	74.1	34.8	55.9	74.3
DETRReg+pseudo-box	71.6	87.0	79.1	36.1	59.0	77.9
Simple Self-training	71.6	87.9	79.7	33.5	60.2	**78.7**
Supervised	**72.6**	**88.0**	**80.7**	**37.6**	**62.6**	78.6

Table 7: Fine-tuning results on PASCAL VOC after pre-training with different methods on Objects365.

#pseudo-box	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
30	**52.0**	**69.6**	**56.7**	**36.1**	**55.9**	65.3
60	51.6	69.1	56.6	34.8	55.4	**65.5**
100	51.5	68.9	56.3	34.9	54.7	65.4

Table 8: Ablation experiments on the number of pseudo-boxes for the DETRReg+pseudo-box method.

Method	Encoder	Decoder	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETRReg+pseudo-box	✓	✓	**52.0**	**69.6**	**56.7**	**36.1**	**55.9**	65.3
		✓	49.4	67.1	53.5	32.0	53.2	63.2
	✓		51.5	**69.6**	56.1	35.4	55.3	**65.5**
Simple Self-training	✓	✓	**52.8**	**70.9**	**57.6**	**37.0**	**56.6**	**67.3**
		✓	50.2	68.2	54.3	32.4	54.1	63.6
	✓		51.8	69.6	56.4	34.9	55.4	66.6

Table 9: Effect of Transformer encoder or decoder pre-training.

of 0.50 or 0.75; and also AP_S, AP_M, AP_L as the AP for small, medium, large bounding boxes.

4.2 Comparison to the State-of-the-art

Table 2 shows the object detection result on COCO validation set of $\mathcal{H}$-Deformable-DETR network pre-trained on Objects365 benchmark with our method in comparison with other state-of-the-art object detection systems. Our Simple Self-training approach significantly boosts the performance of $\mathcal{H}$-Deformable-DETR from 57.9% to 59.3% with fewer training epochs. We expect our approach to achieve better results with bigger batch size and epoch number, for instance, batch size of 256 and epoch of 60 are used for the self-supervised pre-training in Siamese DETR (Huang et al., 2023).

4.3 Results on different DETR architectures

As shown in Table 3 and Table 4, we display the results of DETRReg and Simple Self-training with different DETR architectures on the COCO and PASCAL VOC benchmarks.

vanilla DETR, 50 epochs with Deformable-DETR, and 12 epochs with $\mathcal{H}$-Deformable-DETR, or 24 epochs with $\mathcal{H}$-Deformable-DETR in Section 4.2 for better performance. The learning rate drops at 120/150, 40/50, 11/12, and 20/24 respectively. The batch size for pre-training and fine-tuning are both 16.

Metrics. We measure the object detection accuracy for the top 100 detected bounding boxes. Specifically, we compute AP, AP_{50} and AP_{75} as the average precision when using IoU thresholds across the range of 0.50 to 0.95, and exactly
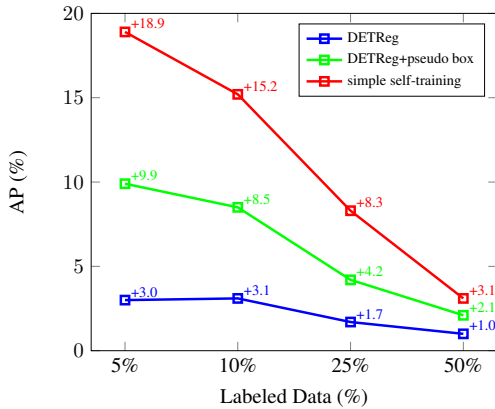
Fig. 5: **Ablation experiments on low-data regimes.** The value shows the performance improvement of three pre-training schemes compared to the *from scratch* baseline.

The line of *from scratch* shows the baseline results without pre-training as the ResNet50 backbone is initialized with SwAV and the Transformer is randomly initialized. The results show that with the reported experiment setting, the DETReg pre-training fails to bring improvement to the *from scratch* baseline on the COCO benchmark, while can get small gains on the PASCAL VOC benchmark. Our Simple Self-training can effectively improve the baseline performance for all three DETR architectures on both benchmarks.

4.4 Ablation Experiments and Analysis

Choice of pre-training dataset. We also investigate the impact of pre-training datasets with the $\mathcal{H}$-Deformable-DETR architecture in Table 5. Compared to ImageNet, pre-training with the Objects365 benchmark yields better performance with the DETReg+pseudo-box and Simple Self-training approach, which is not the case with the DE-TReg approach. As DETReg+pseudo-box and Simple Self-training employ accurate pseudo-boxes of COCO detectors as the pre-training targets, they can benefit from a more complex image scene that contains richer objects like Objects365, while Selective Search's chaotic proposals on Objects365 may not be better localization targets than its proposals on ImageNet. It has been proved that ImageNet is a good benchmark for pre-training general representation ability, which can be transferred to multiple downstream tasks. However, for pre-training a specific detection network, a large-scale object detection benchmark like Objects365 is more helpful if the pseudo-box has good quality. Therefore, we use Objects365 as the default pre-training benchmark for the following studies.

Pre-training methods. We present the downstream results on the COCO benchmark of the *from scratch* and dif-

ferent pre-training methods in Table 6 and results on the PASCAL VOC benchmark in Table 7. All methods except *from scratch* are pre-trained on the Objects365 benchmark. The middle three pre-training methods do not utilize Objects365 ground-truth labels, while the last is supervised by ground-truth and thereby serves as an upper bound. The difference between the three unsupervised pre-training pipelines is illustrated in Figure 3. As shown in the Table 6 and 7, the DETReg+pseudo-box method builds upon DETReg and improves its localization targets by utilizing more accurate COCO detector pseudo-boxes, leading to significant improvement. The Simple Self-training method discards the object-embedding loss and instead supervises the multi-class classification head with the class predictions of COCO detectors, resulting in further performance gains. For the supervised method, we replace the pseudo-box and pseudo-class targets in the Simple Self-training with ground-truth and achieve an upper-bound performance that is only slightly better than our Simple Self-training strategy. This step-by-step comparison demonstrates how we can progressively improve the pre-training performance by introducing better localization and classification targets. Additionally, we observe that better localization pre-training targets are more impactful than better classification targets for object detection tasks.

Pseudo-box Number. In Table 8, we ablate with the number of pseudo-boxes in the DETReg + pseudo-box method. We observe that using more than 30 pseudo-boxes for pre-training does not improve the performance, despite more pseudo-boxes exhibiting higher recall on the ground-truth (as shown in Table 1, where AR@10, 30, 100 means AR with 10, 30, 100 proposed boxes) and providing more supervision signals. A possible explanation is that each Objects365 image contains approximately 15 box annotations, and the predictions beyond the top 30 may have low confidence and less meaningful information, as a result of incorporating noise into the pseudo-box target.

Encoder/Decoder Pre-training. We evaluate the importance of Transformer encoder and decoder pre-training in the DETReg+pseudo-box and Simple Self-training approaches in Table 9. We first report the performance of using both the encoder and decoder pre-trained parameters, then we report the results of only loading the encoder or decoder pre-trained parameters and random initializing the other part. In both pre-training approaches, we observe that the encoder pre-training contributes more than the decoder pre-training, which is reasonable considering the high ratio of encoder GFLOPs shown in 1b.

Fine-tuning dataset size. We investigate the effectiveness of three pre-training schemes compared to training *from scratch* when only a limited amount of data is available for

Text prompt	Gernerative model	Pretraining dataset	Localization target	Classification target	COCO			PASCAL VOC		
					AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
-	-	O365	Pseudo-box prediction	Pseudo-class prediction	52.8	70.9	57.6	71.6	**87.9**	79.7
COCO captions	ControlNet	Control-COCO 2M	Ground-truth	Ground-truth	51.1	69.2	55.8	71.7	87.8	79.2
COCO captions	ControlNet	Control-COCO 2M	Pseudo-box prediction	Pseudo-class prediction	52.6	70.6	57.5	72.0	87.8	**80.4**
LLaVA captions	ControlNet	LLaVAControl-COCO 2M	Ground-truth	Ground-truth	50.7	69.6	55.4	71.6	87.5	79.5
LLaVA captions	ControlNet	LLaVAControl-COCO 2M	Pseudo-box prediction	Pseudo-class prediction	**52.9**	70.8	57.9	**72.3**	87.7	**80.4**
COCO captions	SDXL	SDXL-COCO 2M	Pseudo-box prediction	Pseudo-class prediction	52.5	70.7	57.3	72.1	87.6	79.7
LLaVA captions	SDXL	SDXL-COCO 2M	Pseudo-box prediction	Pseudo-class prediction	**52.9**	**71.0**	**58.0**	72.0	87.6	80.1

Table 10: Evaluation results of pre-training with synthetic images similar to COCO generated by text-to-image generative models ControlNet and SDXL. The text prompts given to the generative models are COCO ground-truth captions (represented as COCO captions) or the generated captions by the large multimodal model LLaVA based on COCO images (represented as LLaVA captions).

Text prompt	Gernerative model	Pretraining dataset	Localization target	Classification target	COCO			PASCAL VOC		
					AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
LLaVA captions	ControlNet	LLaVAControl-O365 2M	Pseudo-box prediction	Pseudo-class prediction	52.4	70.5	57.2	71.8	87.6	79.8
LLaVA captions	SDXL	SDXL-O365 2M	Pseudo-box prediction	Pseudo-class prediction	52.6	70.6	57.6	71.6	87.4	79.3

Table 11: Evaluation results of pre-training with synthetic images similar to Objects365 generated by ControlNet and SDXL. Since Objects365 does not have ground-truth captions, the text prompts given to the generative models are generated captions by LLaVA based on Objects365 images (represented as LLaVA captions).

fine-tuning in Figure 5. Specifically, we fine-tune the pre-trained network on 5%, 10%, 25%, and 50% of the COCO training set and evaluate it on the full COCO validation set. All three pre-training schemes greatly speed up the convergence. We observe that DETReg only yields slightly higher performance than random initialization. The Simple Self-training approach remains the most effective, particularly when only a very small amount of data (5%) is available.

Qualitative analysis. Without fine-tuning, we visualize the discriminability scores (Zong et al., 2022) of the pre-trained encoder in Figure 6 to investigate what the encoder has learned in pre-training. From the figures, we can see that DETReg's feature discriminability is seriously disturbed by the background. However, when we utilize improved localization and classification targets in the DETReg+pseudo-box and Simple Self-training approach, finer details are captured. Notably, the Simple Self-training method demonstrates performance that is almost on par with pre-training using ground-truth.

We also visualize the deformable cross-attention of the pre-trained decoder in Figure 7. The colored dots in the image represent the sampling points from all resolution scales, where the color indicates the attention weights, with a lighter color indicating higher attention. As random initialization shows, the initial key points are sampled radially from the center to the edge. All pre-training methods learn

to scatter the sampling points across the entire object of interest with different patterns, while the Simple Self-training pre-trained decoder can sample key points from an accurate range of objects and distribute attention weight more effectively.

4.5 Results with synthetic data generated by T2I

Last, we investigate the effectiveness of pre-training with synthetic data, which is generated using recent large-scale text-to-image generation models. Specifically, we leverage two representative text-to-image models, ControlNet (Zhang and Agrawala, 2023) and SDXL (Podell et al., 2023), to generate images. These models take original captions from the COCO dataset or captions generated by LLaVA (Liu et al., 2023) as prompts for image synthesis. ControlNet uses predicted depth maps from DPT (Ranftl et al., 2021) as conditional input to generate images that match both the depth maps and captions. On the other hand, SDXL generates images solely based on the provided captions without any additional conditions. We create a synthetic dataset comprising 2.3 Million generated images. Figure 8 displays some examples.

Upon analyzing the images produced by ControlNet, we find that they closely resemble the layout of the original
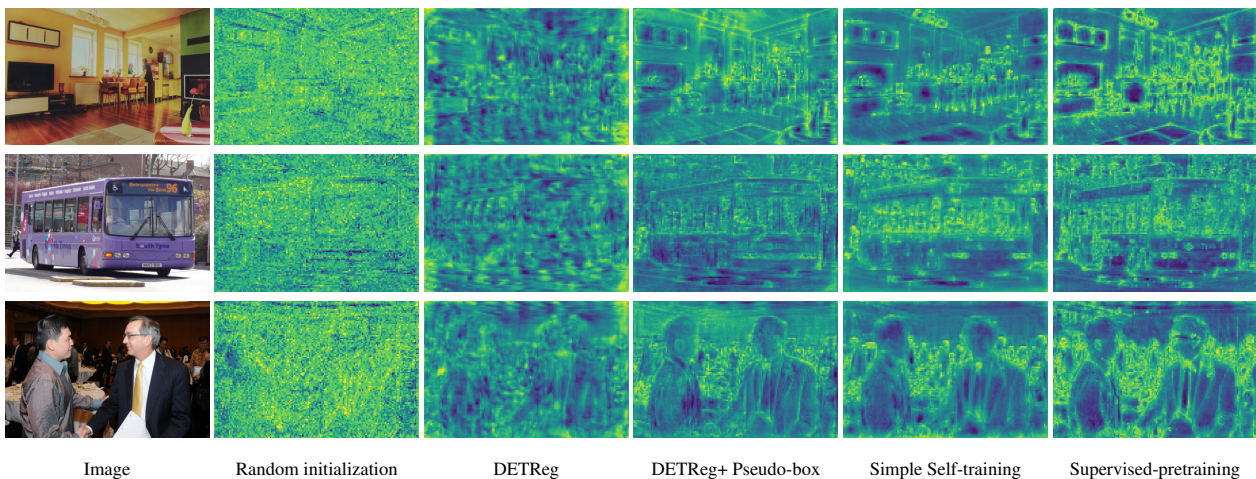
| Image | Random initialization | DETReg | DETReg+ Pseudo-box | Simple Self-training | Supervised-pretraining |

Fig. 6: Visualizations of discriminability scores in the encoder on COCO `val` images.

| Random initialization | DETReg | DETReg+ Pseudo-box | Simple Self-training | Supervised-pretraining |

Fig. 7: Visualizations of deformable cross-attention based on the last Transformer decoder layer on COCO `val` images.

images due to the conditioning on depth maps. This characteristic allows us to use COCO ground-truth data to supervise the pretraining process when using synthetic images generated by ControlNet. Additionally, we also explore the Simple Self-training approach on the synthetic data by pretraining with pseudo-box and pseudo-class predictions that are generated by trained COCO detectors. The process involves pre-training the $\mathcal{H}$-Deformable-DETR model with synthetic images for 3 epochs, followed by fine-tuning on COCO or PASCAL VOC benchmarks for 12 epochs. The results of this evaluation are presented in Table 10. Interestingly, pre-training with the synthetic dataset generated based on COCO demonstrates comparable improvements to pre-training with Objects365 real data using the Simple Self-training scheme. This outcome indicates that text-to-

image synthesis is an effective method for scaling up the original dataset for pre-training. Furthermore, the results on the PASCAL VOC benchmark showcase the generalization ability of pre-training with synthetic data generated based on COCO.

Table 11 shows the results of pre-training with the synthetic data generated based on Objects365 by first captioning Objects365 image with LLaVA and then synthesizing new images from the caption. They are not as good as pretraining with COCO-based synthetic data on both downstream benchmarks.

5 Conclusion

We investigate the effectiveness of DETReg, a representative self-supervised pre-training approach for DETR, across

three distinct DETR architectures. Our findings, unfortunately, do not reveal any performance enhancements of DETReg in recent architectures, thereby challenging the validity of previous conclusions. In response, we reevaluate crucial design aspects, including pre-training targets for localization and classification. As a result of this analysis, we introduce several impactful enhancements and a Simple Self-training scheme that significantly boosts performance across strong DETR architectures. Additionally, we leverage the powerful text-to-image generative models to construct synthetic datasets for pre-training purposes. Remarkably, our approach yields improvements on par with the achievements of pre-training with Objects365. Moving forward, we plan to extend DETR pre-training to encompass a broader spectrum of vision tasks, such as instance segmentation and pose estimation. We hope our work can stimulate the research community to reassess the actual capacity of existing self-supervised pre-training methods when employed in the context of strong DETR models and advance the progress on this challenging task.
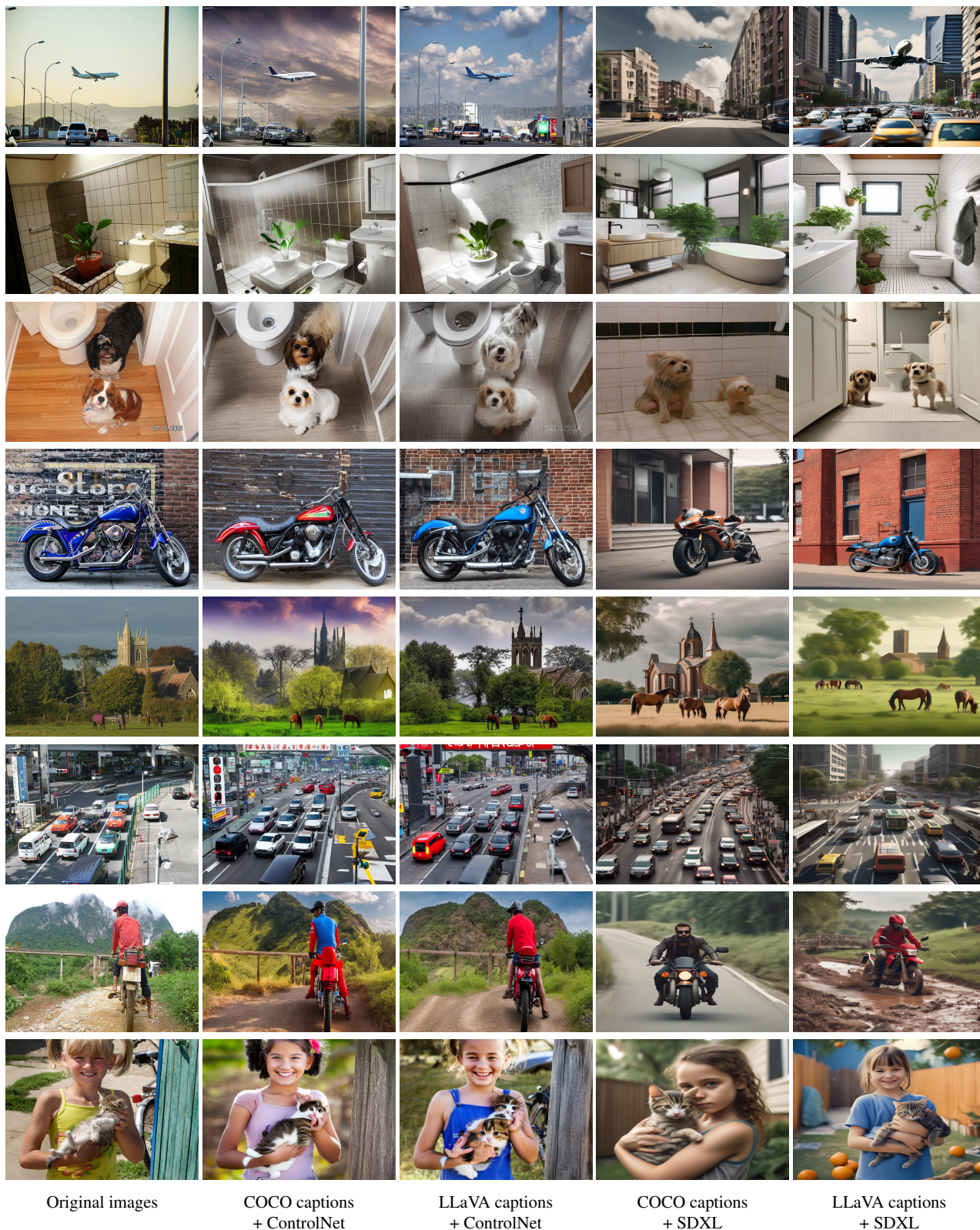
| Original images | COCO captions
+ ControlNet | LLaVA captions
+ ControlNet | COCO captions
+ SDXL | LLaVA captions
+ SDXL |

Fig. 8: Examples of synthetic images using different captions and generative models. The original images are sampled from COCO `train` set.

Data availability statement

The author confirmed that the data supporting the findings of this study are available within the article. Raw data that support the findings of this study and the generated synthetic dataset are available from the corresponding author, upon reasonable request.

References

Bar A, Wang X, Kantorov V, Reed CJ, Herzig R, Chechik G, Rohrbach A, Darrell T, Globerson A (2022) Detreg: Unsupervised pretraining with region priors for object detection. In: CVPR, pp 14605–14615

Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: ECCV, Springer, pp 213–229

Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A (2020) Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS 33:9912–9924

Chen Q, Wang J, Han C, Zhang S, Li Z, Chen X, Chen J, Wang X, Han S, Zhang G, et al. (2022) Group detr v2: Strong object detector with encoder-decoder pretraining. arXiv preprint arXiv:221103594

Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: ICML, PMLR, pp 1597–1607

Dai Z, Cai B, Lin Y, Chen J (2021) Up-detr: Unsupervised pre-training for object detection with transformers. In: CVPR, pp 1601–1610

Grill JB, Strub F, Altché F, Tallec C, Richemond P, Buchatskaya E, Doersch C, Avila Pires B, Guo Z, Gheshlaghi Azar M, et al. (2020) Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS 33:21271–21284

He K, Fan H, Wu Y, Xie S, Girshick R (2020) Momentum contrast for unsupervised visual representation learning. In: CVPR, pp 9729–9738

Huang G, Li W, Teng J, Wang K, Chen Z, Shao J, Loy CC, Sheng L (2023) Siamese detr. In: CVPR, pp 15722–15731

Jia D, Yuan Y, He H, Wu X, Yu H, Lin W, Sun L, Zhang C, Hu H (2023) Detrs with hybrid matching. In: CVPR, pp 19702–19712

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

Li F, Zhang H, Xu H, Liu S, Zhang L, Ni LM, Shum HY (2023) Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In: CVPR, pp 3041–3050

Li Y, Wu CY, Fan H, Mangalam K, Xiong B, Malik J, Feichtenhofer C (2021) Improved multiscale vision transformers for classification and detection. arXiv preprint arXiv:211201526

Li Z, Zhu Y, Yang F, Li W, Zhao C, Chen Y, Chen Z, Xie J, Wu L, Zhao R, et al. (2022) Univip: A unified framework for self-supervised visual pre-training. In: CVPR, pp 14627–14636

Liu H, Li C, Wu Q, Lee YJ (2023) Visual instruction tuning

Liu S, Li F, Zhang H, Yang X, Qi X, Su H, Zhu J, Zhang L (2022a) Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:220112329

Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV, pp 10012–10022

Liu Z, Hu H, Lin Y, Yao Z, Xie Z, Wei Y, Ning J, Cao Y, Zhang Z, Dong L, et al. (2022b) Swin transformer v2: Scaling up capacity and resolution. In: CVPR, pp 12009–12019

Podell D, English Z, Lacey K, Blattmann A, Dockhorn T, Müller J, Penna J, Rombach R (2023) Sdxl: Improving latent diffusion models for high-resolution image synthesis. 2307.01952

Ranftl R, Bochkovskiy A, Koltun V (2021) Vision transformers for dense prediction. In: ICCV, pp 12179–12188

Sahito A, Frank E, Pfahringer B (2022) Better self-training for image classification through self-supervision. In: AJCAI, Springer, pp 645–657

Shao S, Li Z, Zhang T, Peng C, Yu G, Zhang X, Li J, Sun J (2019) Objects365: A large-scale, high-quality dataset for object detection. In: ICCV, pp 8430–8439

Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. IJCV 104(2):154–171

Vandeghen R, Louppe G, Van Droogenbroeck M (2022) Adaptive self-training for object detection. arXiv preprint arXiv:221205911

Wang F, Wang H, Wei C, Yuille A, Shen W (2022) Cp 2: Copy-paste contrastive pretraining for semantic segmentation. In: ECCV, Springer, pp 499–515

Wei F, Gao Y, Wu Z, Hu H, Lin S (2021) Aligning pretraining for detection via object-level contrastive learning. NeurIPS 34:22682–22694

Xie Q, Luong MT, Hovy E, Le QV (2020) Self-training with noisy student improves imagenet classification. In: CVPR, pp 10687–10698

Yang C, Wu Z, Zhou B, Lin S (2021a) Instance localization for self-supervised detection pretraining. In: CVPR, pp 3987–3996

Yang L, Zhuo W, Qi L, Shi Y, Gao Y (2022) St++: Make self-training work better for semi-supervised semantic segmentation. In: CVPR, pp 4268–4277

Yang Q, Wei X, Wang B, Hua XS, Zhang L (2021b) Interactive self-training with mean teachers for semi-supervised object detection. In: CVPR, pp 5941–5950

Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum HY (2022) Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:220303605

Zhang L, Agrawala M (2023) Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:230205543

Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR

Zhu Y, Zhang Z, Wu C, Zhang Z, He T, Zhang H, Manmatha R, Li M, Smola AJ (2021) Improving semantic segmentation via efficient self-training. PAMI

Zong Z, Song G, Liu Y (2022) Detrs with collaborative hybrid assignments training. arXiv preprint arXiv:221112860

Zoph B, Ghiasi G, Lin TY, Cui Y, Liu H, Cubuk ED, Le Q (2020) Rethinking pre-training and self-training. NeurIPS 33:3833–3845