Probing the 3D Object-Level Understanding of Pre-Trained Detection Transformers

Robin Kim[1], Colin Samplawski[2], and Benjamin M. Marlin[1]

[1] UMass Amherst, Amherst, MA, USA
{seunghyunkim,marlin}@cs.umass.edu
[2] SRI International, Menlo Park, CA, USA
colin.samplawski@sri.com

Abstract. Detection transformer models, including DETR and its extensions, learn to output a set of object-level embeddings that can be simultaneously decoded into 2D bounding boxes and class distributions. In this paper, we investigate what pre-trained 2D detection transformers understand about the 3D properties of objects. Specifically, we investigate the extent to which properties including the depth of objects from the camera and the 3D location of objects relative to the camera can be recovered from object-level embeddings using linear and non-linear probes. Across a range of detection transformer models, our results show a surprisingly strong and previously unknown ability of 2D DETR models to represent useful information about the 3D properties of objects, despite the complete lack of 3D supervision during model pre-training.

Keywords: DETR · Probing · Depth Estimation · 3D Understanding

1 Introduction

Detection transformer models including DETR and its extensions [5, 7, 18, 20, 35, 36] are examples of vision transformer models [14] applied to the task of 2D object detection. Detection transformers take individual images as input, and produce as intermediate output a set of embeddings corresponding to latent representations of potential objects in the image. Each query embedding is then decoded into a distribution over classes, and a bounding box in the image plane. Rather than relying on non-maximum suppression (NMS) techniques, the predicted class distribution for each candidate object can be thresholded to determine the final set of predicted objects. Like typical image detection models, the DETR model family is trained using image datasets annotated with classes and 2D bounding boxes [17] only.

The transformer-based encoder-decoder architecture expressed in the original DETR model [5] has been extended to address several other object-oriented, image-based prediction tasks that go beyond predicting object class labels and 2D bounding boxes. Example tasks include instance segmentation [8] and pose estimation [15], as well as monocular 3D object detection [30,34]. Models for such

tasks typically use modified architectures with task-specific decoders or output heads, as well as suitably annotated training data.

In this paper, our primary interest is in the inference of 3D properties of objects from single images. However, unlike past work that has developed and trained models specifically for 3D prediction from images, we pose the question "to what extent do 2D detection transformers encode information about the 3D properties of objects in their latent object embeddings?" Specifically, we investigate the extent to which properties including the depth of objects from the camera and the 3D location of objects relative to the camera can be recovered from pre-trained object-level embeddings using linear and non-linear probes.

Across a range of 2D detection transformer architectures, our results show a surprisingly strong and previously unknown ability of models to infer both depth and 3D location information. On the depth estimation task, we show that the pre-trained latent embeddings of 2D DETR models can be decoded using non-linear probes at a performance level that meets or exceeds that of a zero-shot application of the Depth Anything 2 visual foundation model [32]. On the task of 3D object location prediction, we show that pre-trained latent embeddings of 2D DETR models can be decoded using non-linear probes at an error level that is only fractions of a meter greater than the error of the MonoDETR model, which has an architecture and training approach designed explicitly for this task [34]. Interestingly, our results further show that real-time and lightweight variants of DETR appear to have diminished ability to represent 3D object information compared to other DETR variants.

The remainder of this paper is organized as follows. In Section 2, we discuss related work and describe the models used in our evaluation. In Section 3, we describe our linear and non-linear probing methodology, and define the 3D properties that we investigate. In Section 4, we present experiments and discuss results.

2 Background and Related Work

In this section, we discuss related work and present background information on the models used in our evaluation. We begin with a description of the 2D detection task and the DETR family of models. We then discuss 3D prediction problems and related models. Finally, we discuss related probing studies.

2.1 2D Detection Transformers

The 2D Detection Task. In the standard 2D object detection task, a model is provided with a color input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$. The model issues a set of detections of the form $\mathbf{y} = \{(l_i, \mathbf{b}_i^2) \mid 1 \leq i \leq m\}$ where $l_i \in \mathcal{C}$ is a discrete class label and $\mathbf{b}_i^2$ represents a 2D bounding box in terms of its center $(\mathbf{b}_{x,i}^2, \mathbf{b}_{y,i}^2)$, width $\mathbf{b}_{w,i}^2$, and height $\mathbf{b}_{h,i}^2$. The number of detections to issue m must be predicted by the model. We will refer to the output space of a 2D detector (the set of sets of detections) as $\mathcal{Y}$. To train a standard 2D object detector, a model is

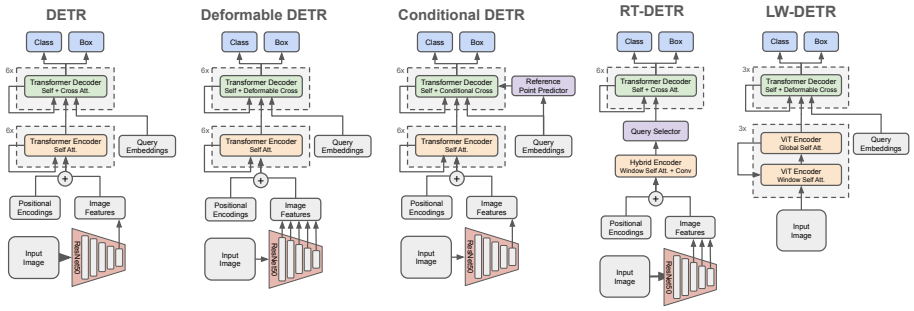

Fig. 1: DETR model variants: DETR, Deformable DETR, Conditional DETR, Real Time DETR, Light Weight DETR.

provided with a dataset $\mathcal{D}_{tr} = \{(\mathbf{x}_n, \mathbf{y}_n) \mid 1 \leq n \leq M\}$ where $\mathbf{x}_n \in \mathbb{R}^{H \times W \times 3}$ is the n^{th} training image and $\mathbf{y}_n = \{(l_{ni}, \mathbf{b}^2_{ni}) \mid 1 \leq i \leq m_n\}$ is the set of ground truth bounding boxes and class labels for the objects present in image n.

The DETR Model. Vision transformers (ViTs) [14] have been applied to many computer vision tasks, including the 2D object detection task described above where they are referred to as detection transformers (DETRs) [5, 7, 18, 20, 35, 36]. DETRs generally follow the high-level architecture of the original DETR model proposed by [5]. This architecture includes a deep backbone f_θ that extracts a pyramid of spatial features. In the original DETR model, the backbone is ResNet50 [13].

These features are then flattened into a sequence of visual tokens that are fed into a transformer encoder e_ϕ consisting of a series of self-attention layers that aggregate information across all visual tokens. The key innovation of the DETR family is the use of a transformer decoder d_ψ that acts on a fixed set of N query embeddings $q \in \mathbb{R}^{N \times d}$. These query embeddings are a set of free parameters that are fed into the decoder along with the visual tokens output by the encoder. The decoder consists of interleaved layers of self-attention on the queries, and cross-attention between the queries and the visual tokens. The original DETR model used transformer layers based on standard scaled dot product attention [28].

The output of the decoder (still $\mathbb{R}^{N \times d}$) is then a set of updated queries which act as a set of object-level embeddings or tokens. Each embedding is then regressed into object-level output representations using a collection of output heads. The original DETR model includes one output head $h^C_{\omega_C}$ that predicts a distribution over the object classes $\mathcal{C}$, and one output head $h^B_{\omega_B}$ that predicts the bounding box coordinates in $\mathbb{R}^4$. In the original DETR model, the class prediction head $h^C_{\omega_C}$ consists of a linear layer followed by a softmax mapping, while the bounding box prediction head is a small MLP.

The DETR model architecture is sketched in Figure 1 along with the additional model variants that we describe in the following sections.

Deformable DETR. A major shortcoming of the original DETR model is that it has poor training efficiency, requiring a large number of training epochs

to achieve reasonable performance. This has been attributed to the standard scaled dot product attention layers used throughout the model, as they lack inductive bias for image-based data. To address this issue, the Deformable DETR model [36] instead uses a deformable attention layer. Here each position in the visual token sequence predicts a small number of other locations to attend to, across all feature scales. This allows for focused, global communication across the image plane, enabling faster convergence during model training.

Conditional DETR. Following a similar logic to Deformable DETR, the Conditional DETR model [20] maintains the dense self-attention of the original DETR model, but adds an additional reference point prediction inside the decoder. Here a small auxiliary network predicts a point in a 2D normalized grid from each initial query embedding. These 2D reference points are then converted into sinusoidal positional embeddings. The positional embeddings are appended onto the query embeddings during the cross-attention operations inside each decoder layer, enabling more efficient localization within the image, and therefore faster training.

RT-DETR. An additional significant issue with the original DETR model is that it is significantly slower at prediction time than detection models using traditional CNN backbones, and explicit non-maximum suppression such as the YOLO family of detectors [25]. This led to several modified DETR architectures that aim to increase detection speed, including the Real-Time DETR (RT-DETR) model [18, 35]. The primary insight in the RT-DETR model architecture is that the simultaneous intra-scale and inter-scale attention used in Deformable DETR is not needed in 2D detection tasks. The model architecture instead uses a CNN to fuse information across scales, and attention to fuse information within a restricted set of scales. In addition, the approach uses a query selection scheme to initialize queries based on initial classification confidence estimates. The approach has been shown to be several times faster than deformable DETR variants, and on par with the run time of recent YOLO model variants [35].

LW-DETR. Light Weight DETR (LW-DETR) is a DETR variant that, like RT-DETR, focuses on prediction speed. While RT-DETR retains a CNN backbone, LW-DETR instead uses a ViT backbone. To deal with the quadratic computation of the ViT backbone, LW-DETR interleaves global attention with windowed attention. The LW-DETR architecture uses a transformer decoder with deformable cross attention. Unlike other DETR architectures, the decoder uses only three layers to further reduce latency.

2.2 3D Vision Transformers

3D Prediction Tasks. Given an input image $\mathbf{x}$, there are multiple 3D prediction tasks of interest. The closest task to the 2D detection task is the 3D detection task, which consists of predicting the class label along with a 3D bounding box for each object in the image. Formally, the output of a 3D detector is given by $\mathbf{y} = \{(l_i, \mathbf{b}_i^3, \phi_i) \mid 1 \leq i \leq m\}$ where $l_i \in \mathcal{C}$ is a discrete class label, $\mathbf{b}_i^3$ specifies the center and extent of a 3D bounding box $\mathbf{b}_i^3 = (\mathbf{b}_{x,i}^3, \mathbf{b}_{y,i}^3, \mathbf{b}_{z,i}^3, \mathbf{b}_{l,i}^3, \mathbf{b}_{w,i}^3, \mathbf{b}_{h,i}^3)$,

and $\phi_i = (\phi_i^y, \phi_i^p, \phi_i^r)$ represents the yaw, pitch and roll angles of the bounding box in the camera's local 3D coordinate system. We note that models developed for specific applications may only include a subset of these outputs.

Another 3D prediction task of interest is inferring a dense depth map given an input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$. The output of this prediction task is a single matrix $\mathbf{y} \in \mathbb{R}^{H \times W}$ giving the depth of the surface represented by each pixel location (i, j) in $\mathbf{x}$. While the depth map prediction task is not an object-oriented task, the output of the depth map prediction task can be combined with the output of a 2D detection model to extract information about the depth of detected objects from the camera.

DETR 3D. The DETR 3D model [30] outputs per-object 3D bounding boxes following a generalization of the 3D detection task described above. Specifically, the input consists of multiple images of the same scene. The model architecture leverages components of the general DETR framework including a ResNet backbone, and transformer self-attention layers. However, the architecture is decoder-only following the backbone, and the main innovation is the use of camera intrinsic and extrinsic information to project queries and features back and forth between 2D and 3D representations. The model is trained using 3D box ground truth.

MonoDETR. The MonoDETR model addresses the 3D prediction task using single image inputs as described above [34]. The MonoDETR architecture uses multi-scale features from a ResNet50 backbone, followed by distinct encoder pathways for RGB image features and depth features. The outputs of the two encoder pathways are fused in the decoder. Like DETR, the decoder uses learned query embeddings, but alternates between attention computations based on image embeddings and depth embeddings. The model is trained on a combination of sparse object-level 2D depth data, and 3D box ground truth. Training and evaluation focus on the KITTI dataset [34].

Depth Anything. The Depth Anything model family addresses the task of predicting dense depth maps from individual images [32]. The model combines a DINOv2 (self-DIstillation with NO labels) vision transformer encoder [22] with a DPT (Dense Prediction Transformer) decoder [24]. The primary innovation in the model is the training procedure, which uses large-scale synthetic image/depth data to train a base model, followed by the use of large-scale real data to drive a distillation process that enables producing accurate lighter-weight models.

2.3 Probing of Deep Models

Due to the poor interpretability of modern deep networks, probing techniques have emerged as a method to infer what concepts are encoded by the latent representation of a pre-trained model [1]. Probing techniques typically train a simple linear or MLP model to predict a property of interest using frozen, pre-trained intermediate features as input. The lower the prediction error of the trained probe model, the greater the degree to which the frozen features encode information about the property of interest. In recent years, these techniques have become a particularly popular choice for interpreting large language models

[2, 19, 23]. Naturally, these techniques have also been applied to recent vision models [6, 11, 16], including (CNN-based) object detectors [21, 29].

More specifically, prior work has investigated whether recent vision foundation models encode 3D and physical scene understanding. For example, [10, 33] studied 3D awareness of pre-trained vision foundation models including DINOv2 [22] and Stable Diffusion [26]. They both conclude that DINOv2 and Stable Diffusion encode some degree of 3D knowledge, even though they are trained without 3D supervision. The DepthCues benchmark of [9] provides a number of tasks which test a probe's ability to understand 3D information based on a single image. These works show that pre-trained visual representations contain significant geometric, physical, and 3D-specific knowledge.

The focus of our work is different from these prior works. They primarily probe generic backbone features, dense feature maps, or pooled region/scene representations. Even when object-level tasks are considered, the unit of analysis is not the detector's query embeddings. Even prior work that explicitly probes object detectors only considers CNN-based detectors [21, 29]. To the best of our knowledge, no prior work has considered probing the query embeddings output by transformer-based detectors. This paper addresses this gap by focusing on the object-level 3D understanding of the DETR family of 2D object detection models.

3 Methods

In this section, we describe the models, 3D properties, and datasets used as the basis for our experiments.

3.1 Models, Query Embeddings and 3D Properties

Our goal in this work is to investigate the extent to which pre-trained 2D detection transformers learn latent representations of 3D properties of objects. We focus on the five DETR model variants described in Section 2: DETR, Deformable DETR, Conditional DETR, RT-DETR v2, and LW-DETR. These models provide a representative sample of 2D detection transformer architectures, exhibiting different choices for the backbone model, the use of multi-scale features, the use of different attention mechanisms, and the use of different encoder and decoder depths. We summarize properties of the models we use in Table 1.

As described in Section 2, all of these models produce a set of embeddings $\{\mathbf{q}_i \in \mathbb{R}^D\}_{i=1}^N$ of dimension D for each of N candidate objects as an intermediate representation. The dimension $D = 256$ for all models considered, except for LW-DETR where $D = 384$. We operationalize our goal of determining the extent to which pre-trained 2D detection transformers learn latent representations of 3D properties of objects in terms of the extent to which 3D properties of objects can be predicted from object embeddings $\mathbf{q}_i$. We address this question for select 3D properties using the probing approach described in the next section. In this work, we focus on two fundamental 3D properties of objects, their depth from the camera and their location in 3D space.

Table 1: Overview of models used in this study. DETR models in the top section of the table are trained without 3D supervision. Depth Anything 2 and MonoDETR are trained using different forms of 3D supervision and serve as baselines for 3D tasks.

Model	Backbone	Supervision	Dataset
DETR [5]	ResNet	Supervised	ImageNet + COCO
LW-DETR [7]	ViT	SSL + Supervised	Object365 + COCO
RT-DETR v2 [18]	ResNet	Supervised	ImageNet + COCO
Conditional DETR [20]	ResNet	Supervised	ImageNet + COCO
Deformable DETR [36]	ResNet	Supervised	ImageNet + COCO
Depth Anything 2 [32]	ViT	SSL + Supervised	Synthetic + Real
MonoDETR [34]	ResNet	Supervised	ImageNet + KITTI3D

3.2 Probing Methodology

In order to assess the extent to which 3D information is encoded in the object embeddings of 2D DETR models, we adopt a probing approach that has been widely used in the analysis of latent representations [1, 3]. Given a collection of K 3D object-level properties P that we would like to jointly assess, we begin by constructing a training dataset $\mathcal{D}_{tr}^{P} = \{(\mathbf{q}_n, \mathbf{p}_n) \mid 1 \leq n \leq M_{tr}\}$ where each $\mathbf{q}_n \in \mathbb{R}^D$ is an object embedding and each $\mathbf{p}_n \in \mathcal{P}$ is the corresponding vector of property values. The set $\mathcal{P}$ represents the space of possible property vectors.

Next, we select a probing model $f_\theta : \mathbb{R}^D \to \mathcal{P}$. In this work, we consider linear probe models f_θ^{lin}, and two-layer fully-connected neural network (MLP) probe models using ReLU activations f_θ^{mlp}. We use the training dataset $\mathcal{D}_{tr}^{P}$ to learn the parameters θ of the probe model f_θ, and then use a held out test dataset of object embeddings and property vectors $\mathcal{D}_{te}^{P} = \{(\mathbf{q}_n, \mathbf{p}_n) \mid 1 \leq n \leq M_{te}\}$ to assess the performance of the learned probe.

For all experiments, the detector weights are frozen and only the probe is trained. The linear probe consists of a single affine map from the detector embedding to the target space. The non-linear probe is a two-layer MLP with ReLU activations and hidden dimension 256. Unless otherwise noted, we use a batch size of 512, learning rate 10^{-3}, and mean squared error as the optimization objective for regression tasks. For the MLP depth experiments, we use 1000 epochs with a 20-epoch warm up followed by cosine decay. We use Adam as the optimizer.

We report predictive performance on the test set in terms of mean absolute error $\text{MAE} = \frac{1}{N}\sum_i |\hat{y}_i - y_i|$, threshold accuracy $\delta_1 = \frac{1}{N}\sum_i \mathbb{I}[\max(\hat{y}_i/y_i,\, y_i/\hat{y}_i) < 1.25]$, and absolute relative error $\text{AbsRel} = \frac{1}{N}\sum_i \frac{|\hat{y}_i - y_i|}{y_i}$ for each property.

The next question is how to construct the necessary datasets for properties of interest given that we need to relate the object embeddings predicted by multiple pre-trained detectors to properties of the predicted objects. We use different approaches for different probe tasks based on the affordances of existing datasets, as described in the next section.

3.3 Probe Dataset Construction

Our general approach to constructing the datasets $\mathcal{D}^P_{tr}$ and $\mathcal{D}^P_{te}$ needed to train and test probe models for different 3D properties leverages base training and test sets $\mathcal{D}^B_{tr} = \{(\mathbf{x}_n, \mathbf{y}_n) \mid 1 \leq n \leq M_{tr}\}$ and $\mathcal{D}^B_{te} = \{(\mathbf{x}_n, \mathbf{y}_n) \mid 1 \leq n \leq M_{te}\}$ where each $\mathbf{x}_n$ is an image and each $\mathbf{y}_n$ is a task-specific annotation structure. For each image $\mathbf{x}_n$ in $\mathcal{D}^B_{tr}$, we run a pre-trained 2D DETR model to obtain the object embeddings $\mathbf{q}_{ni}$, class label probability distributions $\mathbf{l}_{ni}$ and 2D bounding boxes $\mathbf{b}^2_{ni}$. We consider an embedding $\mathbf{q}_{ni}$ to predict the existence of an object if the predicted probability that the object belongs to the set of foreground classes as specified by $\mathbf{l}_{ni}$ exceeds a threshold τ.

For each query embedding $\mathbf{q}_{ni}$ that passes the objectness threshold, we extract the needed property values $\mathbf{p}_{ni}$ from the annotation structure $\mathbf{y}_n$ for training image n. This allows us to form the training pairs $(\mathbf{q}_{ni}, \mathbf{p}_{ni})$ needed to construct $\mathcal{D}^P_{tr}$. We use the same process on the base test dataset $\mathcal{D}^B_{te}$ to form the dataset of test instances $\mathcal{D}^P_{te}$. We provide details for specific property sets below.

Monocular Depth Estimation. For the case of monocular depth estimation, we define the 3D property of interest to be D, the depth from the camera at the center of the predicted bounding box for each predicted object. To build the datasets $\mathcal{D}^D_{tr}$ and $\mathcal{D}^D_{te}$, we assume access to base datasets where each $\mathbf{x}_n$ is an image and each $\mathbf{y}_n$ is a dense depth map (either ground truth, or predicted by a dense depth model). For each query embedding $\mathbf{q}_{ni}$ that passes the objectness threshold, we extract the depth value d_{ni} at the predicted bounding box center location $(\mathbf{b}^2_{x,ni}, \mathbf{b}^2_{y,ni})$ from the dense depth map $\mathbf{y}_n$. This allows us to form the training pairs $(\mathbf{q}_{ni}, d_{ni})$ needed to construct $\mathcal{D}^D_{tr}$. We use the same process on the base test dataset $\mathcal{D}^B_{te}$ to form the dataset of test instances $\mathcal{D}^D_{te}$.

3D Location Prediction. The second task that we consider is 3D location prediction. For this task, the property of interest is the location of the 3D center C of an object in the coordinate system of the camera. To build the datasets $\mathcal{D}^C_{tr}$ and $\mathcal{D}^C_{te}$, we assume access to base datasets where each $\mathbf{x}_n$ is an image and each $\mathbf{y}_n$ is a set of annotations including 2D and 3D bounding boxes for each object present in each image. We extract the set of object embeddings $\mathbf{q}_{ni}$ that passes the objectness threshold for each image. We then compute the IoU of each predicted 2D bounding box $\mathbf{b}^2_{ni}$ with each ground truth bounding box $\mathbf{b}^2_{nj}$. We rank the pairs and perform a greedy matching. Predicted boxes that do not match any ground truth box are discarded. For each object embedding $\mathbf{q}_{ni}$ that is matched with a ground truth object j in image n, we extract the 3D bounding box center $(\mathbf{b}^3_{x,nj}, \mathbf{b}^3_{y,nj}, \mathbf{b}^3_{z,nj})$ as its property vector.

Dataset Alignment. Since the dataset construction method described above is based on filtering detections produced by a given pre-trained DETR model, the approach will result in training and test datasets for different sets of predicted objects. To facilitate comparisons between DETR model variants in some experiments, we align the probing datasets across models after they are constructed using a second level filtering process.

Specifically, we select one DETR model to use as an anchor model. For each image n, we compute the IoU between each 2D bounding box $\mathbf{b}^2_{ni}$ predicted

by the anchor model and each 2D bounding box $\mathbf{b}^2_{nj}$ predicted by an alternate model. We threshold the matches at an IoU of 0.5, rank the surviving pairs, and greedily match them. Lastly, for each predicted object i produced by the anchor model for each image n, we require that it be matched to some prediction j produced by each of the alternate models on the same image. For each model, we discard all instances from its training and test sets $\mathcal{D}^P_{tr}$ and $\mathcal{D}^P_{te}$ that do not satisfy this criterion, resulting in new filtered datasets $\mathcal{F}^P_{tr}$ and $\mathcal{F}^P_{te}$. The filtered datasets include an intersection set of objects that are detected simultaneously by all models.

4 Experiments

In this section, we describe experiments and results assessing the extent to which pre-trained 2D DETR models contain latent representations of 3D properties of objects. As described in Section 3, we focus on monocular depth estimation and 3D object center prediction as probing tasks. The code and experiment configurations are available at `https://github.com/reml-lab/DETRProbe3D/`.

4.1 Monocular Depth Estimation

Experimental Protocol. We evaluate object-level monocular depth understanding using probing datasets constructed from two widely used monocular depth estimation base datasets: Virtual KITTI 2 [4], and NYU Depth v2 (NYUv2) [27]. Virtual KITTI 2 provides synthetic outdoor driving scenes of resolution 1242×375 with dense metric depth, and multiple weather and viewpoint variants. NYUv2 consists of 640×480 real indoor RGBD pairs, allowing us to test depth understanding for indoor scenes. We apply the monocular depth estimation probe dataset construction approach described in Section 3 with an objectness threshold of $\tau = 0.5$. For Virtual KITTI 2, we keep only detections mapped to the `car` class. For NYUv2, we keep all detected classes. We filter out instances with depth values outside of 0.5–100 m for Virtual KITTI 2 and outside of 0.1–10 m for NYUv2 to minimize noise in true depth values.

For this experiment, we apply the dataset alignment process described in Section 3 using DETR as the anchor model. The aligned dataset sizes for Virtual KITTI 2 are ($N_{\text{train}} = 3012$, $N_{\text{test}} = 754$) and for NYUv2 ($N_{\text{train}} = 4600$, $N_{\text{test}} = 1150$). We additionally run Depth Anything 2 on the same images and sample its depth predictions at the anchor model's box centers, producing a center aligned depth baseline. Similarly, we apply MonoDETR both zero-shot, and with a learned MLP output head matching the architecture of the 2D DETR model probes. We use model checkpoints from the Hugging Face library [31]. All detector weights are loaded directly from public Hugging Face checkpoints using `from_pretrained` with model-specific processor classes.

Results Table 2 presents the monocular depth estimation results when applying the probe to the last decoder layer of each DETR model. Overall, the DETR-family latent representations produce competitive depth predictions on

Table 2: Monocular depth estimation results. Best values obtained by probing 2D detection models are shown in **bold**. Best values obtained from 3D models are underlined. These results show that DETR family latent representations contain significant object-level depth information.

		Virtual KITTI 2 (Outdoor)			NYUv2 (Indoor)		
Model	**Probe**	MAE↓	AbsRel (%)↓	δ_1 (%)↑	MAE↓	AbsRel (%)↓	δ_1 (%)↑
DETR	Linear	1.01	8.29	93.10	0.58	24.30	57.22
Conditional-DETR	Linear	1.13	9.07	92.57	0.53	22.52	60.96
Deformable-DETR	Linear	1.06	8.34	91.78	0.56	23.45	59.91
LW-DETR	Linear	1.26	9.84	91.38	0.59	25.30	58.70
RT-DETR v2	Linear	1.56	12.49	86.21	0.64	26.80	55.22
DETR	MLP	**0.54**	**3.62**	98.81	0.51	20.20	65.48
Conditional-DETR	MLP	0.69	4.39	98.67	**0.50**	**19.99**	**66.78**
Deformable-DETR	MLP	0.65	4.05	**98.94**	0.54	21.49	63.65
LW-DETR	MLP	0.88	5.63	98.01	0.55	22.65	62.70
RT-DETR v2	MLP	1.05	6.73	97.21	0.56	22.70	60.43
Depth Anything 2 (Zero-shot)	-	1.49	9.13	97.88	0.61	24.21	59.57
MonoDETR (Zero-shot)	-	3.83	39.82	64.72	10.59	497.72	0.00
MonoDETR (MLP head)	-	0.47	3.14	99.42	0.75	21.49	62.5
Depth Anything 2 (MLP head)	-	0.50	3.30	99.47	0.36	13.36	84.00

both datasets using both linear and MLP probes. We can clearly see that MLP probes outperform linear probes, indicating that the relationship between latent representations and depth estimates is non-linear. We note that among the DETR variants, the real-time models (RT-DETR v2 and LW-DETR) show weaker performance on Virtual KITTI 2 compared to the other three DETR variants. A plausible explanation is that these architectures reduce layers and capacity to meet latency constraints, which may limit the richness of depth cues encoded in the object representations. Despite differences in backbone and attention structures, DETR, Deformable DETR, and Conditional DETR exhibit fairly similar performance.

Further, we see that when Depth Anything 2 and MonoDETR are applied zero-shot on these datasets, they are both out-performed in terms of MAE by all 2D DETR models combined with the MLP probe. While this is surprising as both Depth Anything 2 and MonoDETR are trained with 3D supervision, we note that both models experience domain shift when applied to our specific tasks. To address this, we learn modified output heads for both models with the rest of the parameters frozen to enable fine-tuning to our tasks. For Depth Anything 2, we spatially rescale the patch of output depth values inside the predicted bounding box of each object to a constant size, and learn an MLP output head on this representation. For MonoDETR, we add an MLP output head on top of the last decoder layer representation. Both MLPs have an architecture matching that of the MLP probes used with the 2D DETR models. The results are shown in the bottom two rows of Table 2. We can see that on the Virtual KITTI 2 dataset, both models improve significantly using output heads learned for this task. On the NYUv2 dataset, Depth Anything 2 improves significantly, while MonoDETR is still out-performed by some of the DETR models with MLP probes. The continued weak performance of MonoDETR on NYUv2 is

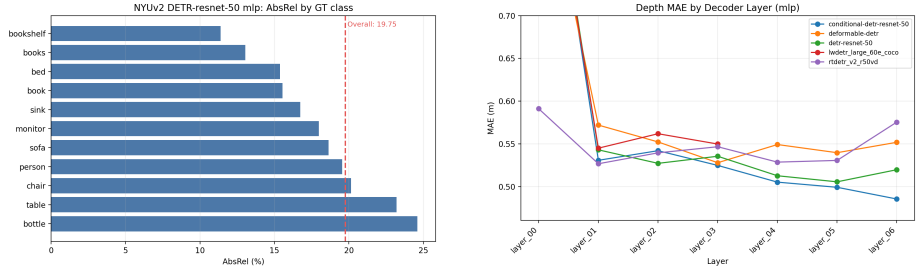

Fig. 2: Further analysis on NYUv2 dataset. (Left) Classwise absolute relative depth prediction error for DETR. (Right) Layerwise depth mean absolute error for all DETR models.

likely attributable to the objects in NYUv2 being out of domain relative to the dataset used to train MonoDETR. Overall, despite the improved performance of Depth Anything 2 and MonoDETR when fine-tuned for the current tasks, the depth estimation performance of the DETR models combined with MLP probes remains surprisingly and objectively strong given the total lack of 3D supervision during latent representation learning.

Next, we consider the question of how depth estimation performance varies by object class. While our evaluation based on the Virtual KITTI 2 dataset is restricted to a single class, the NYUv2 dataset includes multiple classes. In Figure 2(Left), we show how depth estimation error varies across object classes within the NYUv2 dataset for the original DETR model. These results indicate differences in absolute relative error for different classes.

Finally, we investigate the question of whether probing each DETR model at the last decoder layer is optimal. We report depth estimation MAE on the NYUv2 dataset based on MLP probing at different decoder layers in Figure 2 (Right). These results show that the optimal decoder layer to use for depth probing varies by model type, with some models exhibiting modest improvements in depth MAE at lower layers. Since the RT-DETR decoder initializes the query using tokens from the encoder, even probes on the earliest decoder layer show competitive performance.

4.2 3D Location Prediction

Experimental Protocol. We evaluate 3D object location understanding using a probing dataset constructed from the widely-adopted KITTI [12] benchmark, following the data preprocessing protocol from [34]. We follow the 3D location prediction dataset construction approach described in Section 3 with an objectness threshold of $\tau = 0.5$. For this experiment, we report results for MLP probes only as linear probes again perform poorly. We do not align the datasets for each model to each other as they are already aligned against ground truth objects. We report the MAE metric for each dimension of the object center separately, as well as the MAE between the ground truth 3D object center and the predicted

Table 3: 3D location prediction results. Error is reported for each component of the object center separately, as well as for the full vector of object center coordinates. Center AbsRel is reported as a percentage. Best values obtained by probing 2D detection models are shown in **bold**. Best values obtained from 3D models are underlined.

Model	xMAE (m) ↓	yMAE (m) ↓	zMAE (m) ↓	Center MAE (m) ↓	Center AbsRel (%) ↓
DETR	**0.33**	**0.12**	**1.03**	**0.50**	**2.40**
Conditional-DETR	0.68	0.16	1.09	0.64	3.01
Deformable-DETR	0.66	0.16	1.04	0.62	2.90
RT-DETR v2	2.27	0.18	1.39	1.28	6.49
LW-DETR	2.28	0.17	1.20	1.22	6.01
MonoDETR	0.19	0.06	0.64	0.30	1.41

center. We note that we use the definition of the 3D bounding box center used in the KITTI dataset, which corresponds to the bottom center of each 3D box.

Results Table 3 reports 3D object location results for all of the 2D DETR models as well as for MonoDETR. Among the 2D DETR family models, DETR achieves the best performance. Deformable-DETR and Conditional-DETR have similar performance, while the real-time variants are again noticeably worse. While MonoDETR achieves the best performance on this task due to learning all model parameters using full 3D bounding box ground truth on this task, the results for the best performing DETR models with MLP probes are again surprisingly strong. The overall MAE for DETR is 0.5m compared to MonoDETR's 0.3m. We emphasize that the 0.2m MAE difference between the best DETR MLP probe and MonoDETR on KITTI 3D is quite small. The median distance to the objects in our test dataset is 25.3m. As we can see in Table 3, the gap in absolute relative error between the methods is about 1%. This result again suggests that DETR's latent object representations encode significant 3D information despite the total lack of 3D supervision during latent representation learning.

4.3 Further Analysis

Compressibility of Embeddings. In this experiment, we investigate the extent to which object embeddings in the last decoder layer can be linearly compressed while retaining the ability to decode 3D information. We focus on the monocular depth estimation task. We first extract query embeddings, then compress them from D to $k < D$ dimensions for values of k from 2^0 to the original representation size using PCA. Figure 3 shows the results. Across all DETR family models, performance improves rapidly as the PCA dimension increases, and then saturates beyond a moderate dimensionality. This trend suggests that depth-relevant information is concentrated in a relatively low-dimensional subspace of the embedding space. We also observe that real-time variants tend to require more retained dimensions to reach comparable performance, which is consistent with earlier observations regarding the under-performance of the representations produced by these models.

Table 4: Monocular depth estimation with ablation on probe input and probe capacity. We compare embedding probes vs. bounding box probes on the same Virtual KITTI 2/NYUv2 experimental protocol.

	Virtual KITTI 2			NYUv2		
Method	MAE↓	AbsRel (%)↓	δ_1↑	MAE↓	AbsRel (%)↓	δ_1↑
DETR Bbox probe (Linear)	4.27	32.67	44.69	0.87	37.61	41.74
DETR Embedding probe (Linear)	**1.01**	**8.29**	**93.10**	**0.58**	**24.30**	**57.22**
DETR Bbox probe (MLP)	1.02	7.09	95.76	0.77	32.98	51.83
DETR Embedding probe (MLP)	**0.54**	**3.62**	**98.81**	**0.51**	**20.20**	**65.48**

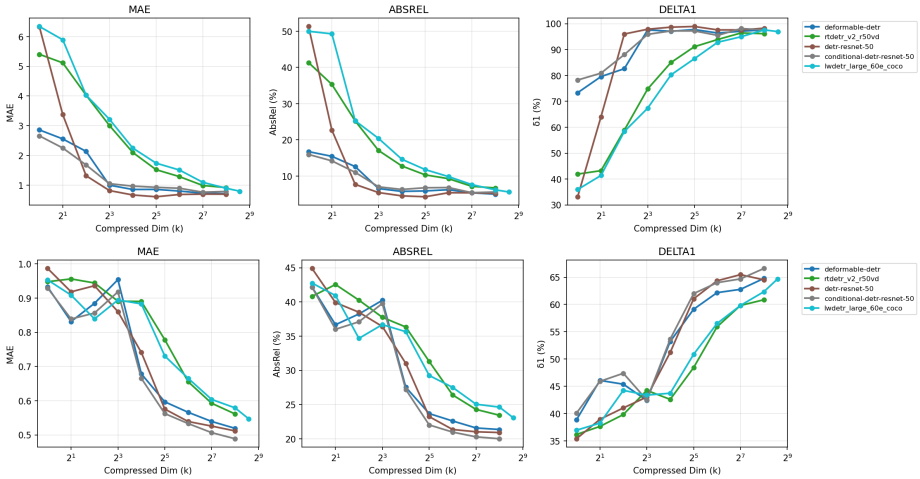

Fig. 3: Monocular depth estimation with MLP probes under PCA compression. Top: Virtual KITTI 2. Bottom: NYUv2. We see rapid performance saturation as k increases, indicating depth information is concentrated in a relatively low-dimensional subspace.

Latent Representation Ablation. In this experiment, we examine the predictive lift of object embeddings compared to 2D bounding box representations. We construct probing datasets for the 2D bounding box case exactly as for query embeddings except that we substitute the predicted bounding box representation $\mathbf{b}$ for the query embedding $\mathbf{q}$. Table 4 shows the results for both linear and MLP probes using the DETR model on the monocular depth estimation task. We provide results on both the Virtual KITTI 2 and NYUv2 datasets. Across both linear and MLP cases, the latent embedding probes dramatically outperform bounding box probes. This consistent margin shows that query embeddings contain substantial depth-relevant information beyond what is decoded into the 2D bounding box representation, likely capturing additional cues such as semantic context, category priors, and other high-level signals that correlate with object depth.

5 Conclusions

In this paper, we study the extent to which 2D DETR family models that are trained without explicit 3D supervision encode object-level 3D properties in their latent object embeddings. We establish methods for constructing probe datasets to assess the understanding of 3D object-level properties, and apply the methodology to the tasks of object-centric monocular depth estimation and 3D object localization. Our results on monocular depth estimation show that 2D DETR models learn representations that can out-perform a zero-shot application of Depth Anything 2 on some scene types. Further, our results on 3D object localization show that 2D DETR models learn representations that can be decoded into high-quality location estimates with fractions of a meter more error than MonoDETR, a model trained specifically for this task using full 3D bounding box supervision. Taken together, these results suggest that 2D DETR models learn latent representations that encode a surprising amount of 3D structure.

Future work includes assessing performance with respect to additional 3D properties, delving deeper into the representation gap between the light weight and real time DETR model variants and the more standard model variants, and leveraging these insights to inform the development of novel 3D DETR model architectures.

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-2-0196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. arXiv preprint arXiv:1610.01644 (2016)
2. Azaria, A., Mitchell, T.: The internal state of an llm knows when it's lying. In: Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing 2023. pp. 967–976 (2023)
3. Belinkov, Y.: Probing classifiers: Promises, shortcomings, and advances. Computational Linguistics **48**(1), 207–219 (2022)
4. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)

6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
7. Chen, Q., Su, X., Zhang, X., Wang, J., Chen, J., Shen, Y., Han, C., Chen, Z., Xu, W., Li, F., et al.: Lw-detr: A transformer replacement to yolo for real-time detection. arXiv preprint arXiv:2406.03459 (2024)
8. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 17864–17875. Curran Associates, Inc. (2021)
9. Danier, D., Aygün, M., Li, C., Bilen, H., Mac Aodha, O.: Depthcues: Evaluating monocular depth perception in large vision models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 20049–20059 (2025)
10. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3d awareness of visual foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21795–21806 (2024)
11. Gairola, S., Böhle, M., Locatello, F., Schiele, B.: How to probe: Simple yet effective techniques for improving post-hoc explanations. In: International Conference on Learning Representations (2025)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
14. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM Computing Surveys **54**(10s), 1–41 (2022)
15. Li, Y., Zhang, S., Wang, Z., Yang, S., Yang, W., Xia, S.T., Zhou, E.: Tokenpose: Learning keypoint tokens for human pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11313–11322 (October 2021)
16. Li, Y., Bornschein, J., Hutter, M.: Evaluating representations with readout model switching. In: International Conference on Learning Representations (2023)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
18. Lv, W., Zhao, Y., Chang, Q., Huang, K., Wang, G., Liu, Y.: Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv preprint arXiv:2407.17140 (2024)
19. Marks, S., Tegmark, M.: The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. arXiv preprint arXiv:2310.06824 (2023)
20. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660 (2021)
21. Mikriukov, G., Schwalbe, G., Hellert, C., Bade, K.: Evaluating the stability of semantic concept representations in cnns for robust explainability. In: World Conference on Explainable Artificial Intelligence. pp. 499–524. Springer (2023)

22. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024)
23. Park, K., Choe, Y.J., Veitch, V.: The linear representation hypothesis and the geometry of large language models. In: Proceedings of the 41st International Conference on Machine Learning. pp. 39643–39666 (2024)
24. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12179–12188 (2021)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
27. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision. pp. 746–760. Springer (2012)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
29. Wagner, B., Pellerin, D., Huet, S.: Forgetting analysis by module probing for online object detection with faster r-cnn. In: 2024 32nd European Signal Processing Conference. pp. 576–580. IEEE (2024)
30. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
31. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45 (2020)
32. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)
33. Zhan, G., Zheng, C., Xie, W., Zisserman, A.: A general protocol to probe large vision models for 3d physical understanding. Advances in Neural Information Processing Systems **37**, 43468–43498 (2024)
34. Zhang, R., Qiu, H., Wang, T., Guo, Z., Cui, Z., Qiao, Y., Li, H., Gao, P.: Monodetr: Depth-guided transformer for monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9155–9166 (2023)
35. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16965–16974 (2024)
36. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)