KeypointDETR: An End-to-End 3D Keypoint Detector

Hairong Jin[1,2⋆], Yuefan Shen[1], Jianwen Lou[3], Kun Zhou[1], and Youyi Zheng[1(✉)]

[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
hairong_jin,jhonve,kunzhou,youyizheng@zju.edu.cn
[2] Chohotech Co.,Ltd., Hangzhou, China
[3] School of Software Technology, Zhejiang University, Hangzhou, China
jianwen.lou@zju.edu.cn

Abstract. 3D keypoint detection plays a pivotal role in 3D shape analysis. The majority of prevalent methods depend on producing a shared heatmap. This approach necessitates subsequent post-processing techniques such as clustering or non-maximum suppression (NMS) to pinpoint keypoints within high-confidence regions, resulting in performance inefficiencies. To address this issue, we introduce KeypointDETR, an end-to-end 3D keypoint detection framework. KeypointDETR is predominantly trained with a bipartite matching loss, which compels the network to forecast sets of heatmaps and probabilities for potential keypoints. Each heatmap highlights one keypoint's location, and the associated probability indicates not only the presence of that specific keypoint but also its semantic consistency. Together with the bipartite matching loss, we utilize a transformer-based network architecture, which incorporates both point-wise and query-wise self-attention within the encoder and decoder, respectively. The point-wise encoder leverages the self-attention mechanism on a dynamic graph derived from the local feature space of each point, resulting in the generation of heatmap features. As a key part of our framework, the query-wise decoder facilitates inter-query information exchange. It captures the underlying connections among keypoints' heatmaps, positions, and semantic attributes via the cross-attention mechanism, enabling the prediction of heatmaps and probabilities. Extensive experiments conducted on the KeypointNet dataset reveal that KeypointDETR outperforms competitive baselines, demonstrating superior performance in keypoint saliency and correspondence estimation tasks. (The code will be released at github.com/bibi547/KeypointDETR)

Keywords: 3D keypoint · Point cloud · Heatmap

1 Introduction

The automatic extraction of 3D keypoints presents a significant challenge in the realms of computer vision and computer graphics. These keypoints are instrumental in various geometric processing tasks, such as object detection [20], shape

⋆ This work was done during the author's internship at Chohotech Co.,Ltd..

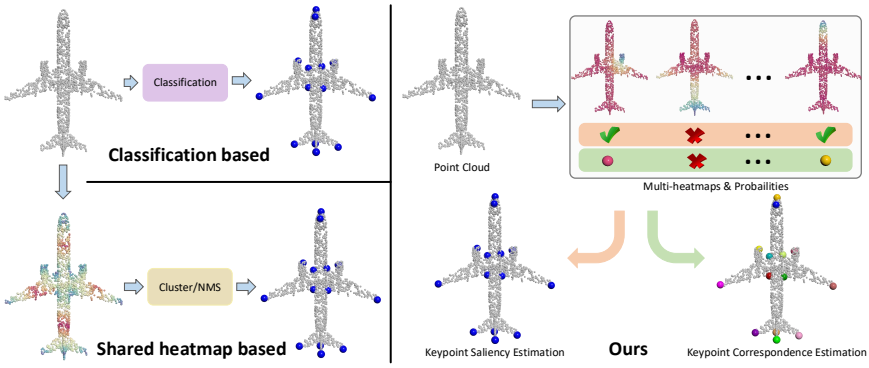


Fig. 1: Illustration of the differences between the existing methods and KeypointDETR for keypoint saliency and correspondence estimation. Given a 3D model, KeypointDETR can predict its multi-heatmaps and probabilities. Heatmaps provide accurate inference of keypoints' positions. Probabilities indicate whether the specific keypoints exist in the 3D model.

segmentation [13, 21], shape matching [39, 44], point registration [1], and point cloud completion [36], *etc*.

Recent breakthroughs in deep learning, especially in 3D shape analysis and geometric feature extraction [5,11,27,28,40,43], have catalyzed the emergence of a multitude of learning-based approaches for 3D keypoint detection. A straightforward technique [45,47] is point classification on the surface of a 3D shape, but this method grapples with significant sample imbalances. To mitigate this, many existing keypoint detection strategies [32,41,42] utilize Gaussian kernel functions to create a ground truth heatmap. This shared heatmap is generated based on the distances between points and keypoints, guiding the extraction of keypoints from areas identified as salient. However, these techniques typically require post-processing steps such as clustering [29] or non-maximum suppression [22] to sift out high-confidence points and finalize the keypoint extraction. These processes often involve computing distance matrices, which can considerably slow down inference speeds. Additionally, the selection of parameters can profoundly affect the accuracy of keypoint extraction. Many landmark detection methods for 2D or 3D images [4] prefer using multi-heatmaps for keypoints, as opposed to relying on a single shared heatmap. This approach assigns each keypoint its unique heatmap, facilitating the extraction of the point with the highest value from each map. Although their strategy eliminates the need for post-processing, it can limit the algorithm's generalizability. One significant challenge is ensuring a consistent number of keypoints across all 3D models within a category.

To tackle these challenges, we introduce an innovative end-to-end 3D keypoint detection model named KeypointDETR. This approach not only eliminates the need for any post-processing steps but also exhibits remarkable generalization capabilities. As illustrated in Fig. 1, our framework takes point coordinates as input and predicts multi-heatmaps along with their associated probabilities,

providing a consistent paradigm for keypoint saliency and correspondence estimation. For keypoint saliency estimation, we can derive binary *yes*/*no* labels from the probabilities, while for keypoint correspondence estimation, semantic labels can be inferred. Our pipeline utilizes a loss function based on the Hungarian algorithm [15], which facilitates the optimal bipartite matching between the predictions and the ground truth heatmaps. Consistent with this strategy, we develop a transformer-based encoder-decoder network architecture. The encoder employs dynamic graph transformer modules to extract geometric features from 3D models, specifically focusing on the generation of heatmap features. We regard the learnable heatmap features as intrinsic geometric features, eliminating the need for direct supervision. The dynamic graph transformer module aggregates the features from neighboring points within the feature space and assigns self-attention weights, effectively extending the range of perception by stacking these modules. In the decoder, we introduce a transformer refinement strategy that utilizes the query-wise self-attention mechanism to facilitate information exchange among potential keypoints. Additionally, it incorporates the cross-attention mechanism to capture the latent connections among keypoints' heatmaps, positions, and semantics. The query-wise decoder optimizes the heatmap and query features through the analysis of inter-keypoint and inter-attribute relationships, thereby enabling the parallel prediction of multi-heatmaps and probabilities.

We validate our method on the KeypointNet dataset [47], achieving outstanding results in two essential tasks: keypoint saliency and correspondence estimation. Furthermore, our approach of assigning a specific heatmap to each keypoint leads to significantly higher accuracy compared to methods that use a shared heatmap.

Our main contributions can be summarized as follows:

- We introduce KeypointDETR, an end-to-end 3D keypoint detection framework. It is trained with a loss function based on the optimal bipartite matching derived from the cost matrix of multi-heatmaps and probabilities.
- We construct a novel transformer-based network architecture. The point-wise encoder harnesses self-attention on the dynamic graph of points' local feature space to extract geometric features. The query-wise decoder incorporates global self-attention to enhance the interaction between queries.
- We conduct a comprehensive evaluation that validates our method and provides insights into the impact of principal components, such as the bipartite matching loss and query-wise self-attention.

2 Related Work

2.1 3D Keypoint Detection

In previous studies on 3D keypoint detection [23, 33, 37], algorithms heavily rely on a variety of hand-crafted geometric features, such as ISS [30], HKS [34],

CGF [14], Salient Points [3], and Mesh Saliency [16]. These traditional approaches primarily focus on local geometric features of 3D shapes, often overlooking the integration of global and semantic information.

With advancements in deep learning, there has been a notable surge in the development of data-driven algorithms for keypoint detection [10, 35, 54]. Numerous unsupervised 3D keypoint detection algorithms [1, 7, 12, 18, 31, 50, 53] are proposed for specific tasks such as point registration and reconstruction. For example, UKPGAN [46] utilizes a GAN-based module to manage keypoint sparsity and reconstructs 3D models by distilling saliency information. Unsupervised methods tend to focus on identifying keypoints in areas with distinct geometric features. While some of these methods are capable of attaining semantic consistency, they typically require the predetermination of the number of keypoints. Furthermore, in many scenarios, it is essential to identify keypoints that may not have prominent geometric features but possess solid semantic importance. Methods like SyncSpecCNN [45] and PRA-Net [5] address keypoint detection by classifying points on 3D shapes. These methods often result in an imbalance between positive and negative samples, presenting a significant challenge. Consequently, current popular methods employ a heatmap-based approach [32, 41], where the value assigned to each point indicates its proximity to the nearest keypoint. Wei *et al*. [42] proposes a multi-task learning framework that combines point-to-keypoint offsets with a confidence map, facilitating effective 3D keypoint saliency and correspondence estimation. Heatmap-based methods for 3D shapes predominantly depend on a shared heatmap for all keypoints and require post-processing. These methods employ post-processing techniques like non-maximum suppression [22] or clustering [29] to isolate keypoints from regions with high scores during the inference process.

2.2 Transformer in Vision

Transformer [38] was initially introduced in the realm of natural language processing. Recent advancements in visual research [6, 19] have explored various models centered around self-attention [38], achieving unprecedented performance. DETR [2] pioneers an innovative end-to-end object detection approach. It formulates a loss matching rule employing the Hungarian algorithm [15], facilitating bipartite matching between predictions and ground truth. This method serves as the cornerstone for a series of object detection methods [4,17,55] based on set prediction. Li *et al*. [17] develop a facial landmark detection method using cascaded transformer decoders. They devise parallel decoders utilizing deformable self-attention [55] to optimize facial landmark coordinates through offset predictions. Chen *et al*. [4] leverage self-attention to create a structure-aware LSTM framework tailored for predicting 3D heatmaps in CBCT. Keypoint transformer [9] efficiently extracts all potential keypoints within the image from a shared heatmap, subsequently employing the DETR decoder to predict 3D hand poses. Given the transformer's inherent capacity to handle input element permutations in sequences, it is exceptionally well-suited for point cloud processing [8, 24, 26].

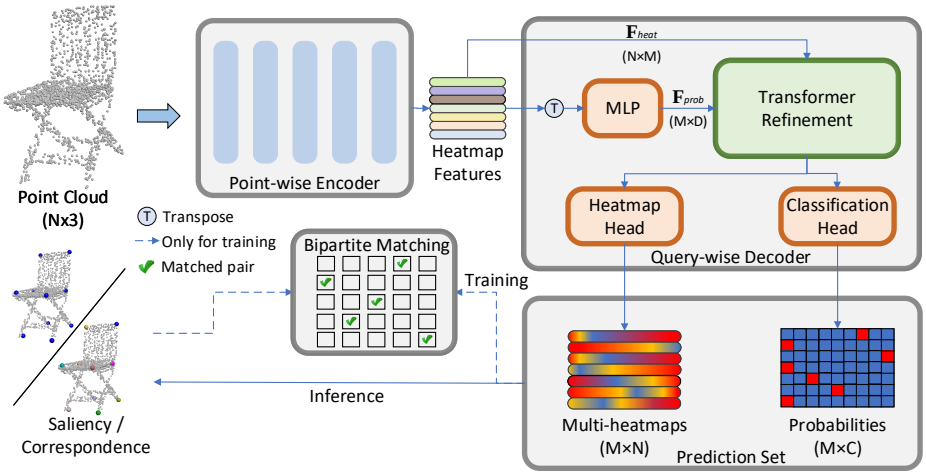
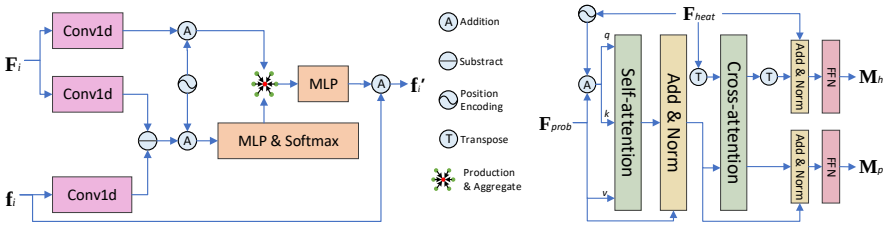


Fig. 2: Pipeline of KeypointDETR for keypoint saliency and correspondence estimation. Our framework takes point coordinates as input. The point-wise encoder first extracts geometric features from the point cloud by stacking dynamic graph transformer modules (DGT). Then, the query-wise decoder accepts heatmap features, which are transformed into heatmap and query features to generate multi-heatmaps and probabilities. These predictions enable the direct inference of both keypoints positions and semantic labels.

Nonetheless, applying global self-attention to 3D point cloud models incurs substantial computational costs. Consequently, existing research [48, 52] frequently adopts local self-attention mechanisms, expanding receptive fields through sampling and grouping strategies. In contrast, our point-wise encoder integrates local self-attention within the dynamic graph of the point feature space, thereby eliminating the necessity for sampling and grouping operations. Distinct from most existing DETR style works, our query-wise decoder is meticulously engineered to predict multi-heatmaps and probabilities, enabling efficient end-to-end 3D keypoint detection.

3 Method

In this section, we delve into the specifics of our end-to-end 3D keypoint detector, KeypointDETR, which consists of two essential components: a network architecture based on the transformer and a loss function formulated using bipartite matching. As illustrated in Fig. 2, our approach is applied to tasks involving keypoint saliency and correspondence estimation. Upon receiving a 3D point cloud as input, our point-wise encoder utilizes the dynamic graph transformer modules to extract heatmap features, which can be considered intrinsic geometric features. We then deploy a query-wise transformer decoder designed to enhance information exchange between queries and capture the latent relationships between attributes through cross-attention, thereby optimizing heatmap



(a) The structure of the dynamic graph transformer module.

(b) The structure of the transformer refinement module.

Fig. 3: The structure of the dynamic graph transformer module (DGT) and transformer refinement module.

and query features. This network yields heatmaps and probabilities as its outputs. The prediction sets are aligned with ground truth heatmaps through a cost matrix, leading to the computation of the final loss based on optimal bipartite matching. During the inference stage, we select heatmaps corresponding to keypoints, as indicated by the predicted probabilities, and extract keypoints exhibiting the highest heat values from these maps.

3.1 Data Preparation

To tackle the issue of imbalanced samples concerning keypoints on the surface of 3D models, heatmap-based 3D keypoint detection methods establish the ground truth heat value for each point based on its distances to the keypoints. Specifically, the heatmap is generated by applying a Gaussian kernel in conjunction with the geodesic distance matrix:

$$h_{i,j} = \exp^{-\frac{1}{2\sigma^2}\mathrm{d}(p_i,k_j)}. \tag{1}$$

Given a 3D point cloud with N_p points $P = \{p_i\}_{i=1}^{N_p}$ and N_k keypoints $K = \{k_j\}_{j=1}^{N_k}$, our process begins with the computation of the geodesic distance matrix from each point to the keypoints. Then, we derive the heatmap matrix $\mathcal{H} = \{h_{i,j}\} \in \mathbb{R}^{N_p \times N_k}$ using the Gaussian kernel function. Contrasting with methods that use a shared heatmap, where the maximum value for each point is considered, our approach distinctively employs multi-heatmaps, each corresponding to specific keypoints, as the ground truth.

3.2 Network Architecture

As shown in Fig. 2, our transformer-based encoder-decoder network architecture primarily comprises a point-wise encoder and a query-wise decoder.

Point-wise Encoder The main objective of the encoder is to extract geometric features from the 3D point cloud. Drawing inspiration from DGCNN [40] and

the Point Transformer [52], we devise a point-wise encoder constructed using dynamic graph transformer modules (DGT).

As depicted in Fig. 3 (a), we employ local self-attention within the dynamic graph transformer module to operate on the feature space. To be specific, for the feature map $\mathbf{F}$ at each layer, we extract a k-nearest neighbors feature set $\mathbf{F}_i = \{\mathbf{f}_j\}_{j=1}^{k}$ for $\mathbf{f}_i$. Thereafter, we assign self-attention weights to these k-nn features and aggregate them using the following equation:

$$\mathbf{f}_i' = \sum_{\mathbf{f}_j \in \mathbf{F}_i} \rho(\theta(\phi(\mathbf{f}_i) - \varphi(\mathbf{f}_j) + \delta)) \odot (\psi(\mathbf{f}_j) + \delta). \tag{2}$$

Here, θ is implemented as an MLP, while ϕ, φ, ψ are realized using Conv1d. ρ represents the softmax operation applied to self-attention weights, and $\odot$ denotes the element-wise product operation. The term δ represents the position encoding, which is generated from the point coordinates $P_i = \{p_j\}_{j=1}^{k}$ corresponding to the neighborhood features, as expressed in the following:

$$\delta = \mathrm{MLP}(p_i - p_j). \tag{3}$$

Finally, we concatenate the features extracted from each dynamic graph transformer module to obtain point-wise features, which are decoded into heatmap features through MLPs.

Query-wise Decoder The decoder's output comprises a predefined number of predictions, referred to as keypoint queries, with the query count M set significantly higher than the number of keypoints within a 3D model. Each prediction query consists of both a heatmap and a probability. Heatmaps serve to indicate the position of keypoints within the 3D model. Probabilities reflect the semantic consistency of potential keypoints. In keypoint saliency estimation, they determine the presence or absence of specific keypoints. In keypoint correspondence estimation, such probabilities are correlated with their respective semantic labels.

Guided by these objectives, we develop a query-wise decoder that adheres to the proven DETR-style architecture [2], as illustrated in Fig. 3 (b). Initially, we take the heatmap features $\mathbf{F}_{heat} \in \mathbb{R}^{N \times M}$ from the encoder. Considering the inherent relationship between the heatmaps and semantics of keypoints, the heatmap features are directly transposed and then decoded into query features $\mathbf{F}_{prob} \in \mathbb{R}^{M \times 128}$ through an MLP. Following this, both the heatmap $\mathbf{F}_{heat}$ and query $\mathbf{F}_{prob}$ features are fed into the transformer decoder. To facilitate the exchange of information among queries, a multi-head self-attention module is applied to $\mathbf{F}_{prob}$. For the position encoding, we extract the coordinates of the reference points corresponding to the peaks of heat values in $\mathbf{F}_{heat}$, which are subsequently processed through an MLP to produce the position encoding. The transposed $\mathbf{F}_{heat}$ and $\mathbf{F}_{prob}$ serve as inputs to the cross-attention module, enhancing the interplay of information between probability and heatmap features. Ultimately, through the application of Feed-Forward Network modules (FFN),

the refined $\mathbf{F}_{heat}$ and $\mathbf{F}_{prob}$ are decoded into multi-heatmaps $\mathbf{M}_h$ and probabilities $\mathbf{M}_p$, respectively.

The decoder utilizes a query-wise self-attention mechanism across the queries, leading to M distinct predictions. This mechanism operates as a global self-attention centered around potential keypoints. Such a globally parallel decoding approach efficiently captures the interactions among the predicted queries, thereby optimizing heat values and reducing redundancy in the identification of keypoints. Additionally, considering the potential relationship between keypoint heatmaps and semantics, cross-attention is employed to facilitate the interaction between heatmap and query features.

3.3 Loss Function

Inspired by DETR [2], we integrate the bipartite matching loss into 3D keypoint detection to enable end-to-end inference, eliminating the need for post-processing. During the bipartite matching phase, we calculate the pairwise matching costs by assessing the discrepancies between the predicted queries and ground truth. We utilize this cost matrix to consolidate the prediction sets, distinguishing between "background" queries and keypoint queries, in order to compute the final loss.

Given a 3D model with N_k ground truth keypoints, KeypointDETR generates a set of M predictions, where M is preconfigured to be significantly larger than N_k. The ground truth can be represented as $\{y_i\}_{i=1}^{N_k} = \{h_i, p_i\}_{i=1}^{N_k}$, where h_i signifies the ground truth heatmap and p_i denotes the ground truth probability. The predicted queries are denoted as $\{\hat{y_j}\}_{j=1}^{M} = \{\hat{h_j}, \hat{p_j}\}_{j=1}^{M}$, with $\hat{h_j}$ representing the predicted heatmap, and $\hat{p_j}$ corresponding to the predicted probabilities associated with keypoints categories $\{c_i\}_{i=1}^{N_k}$. In the task of keypoint saliency estimation, p_i remains fixed at 1, while $\hat{p_j}(c_i)$ signifies the probability of being predicted as a keypoint. However, in the task of keypoint correspondence estimation, $\hat{p_j}(c_i)$ is defined as the probability of c_i, where c_i denotes the semantic label corresponding to the ground truth keypoint k_i. Expanding upon these notations, in order to achieve the optimal bipartite matching between ground truth and predictions, it is imperative to establish a matching cost matrix $\mathcal{M}_c \in \mathbb{R}^{N_k \times M}$. This cost matrix takes into account both the heatmaps and probabilities. Precisely, each element $\mathcal{M}_c(i,j)$ of the cost matrix can be calculated as follows:

$$\mathcal{M}_c(i,j) = \mathbf{W_h} \cdot \sum \mid h_i - \hat{h_j} \mid + \mathbf{W_p} \cdot (1 - \log \hat{p_j}(c_i)), \tag{4}$$

where $\mathbf{W_h}$ represents the weight assigned to heatmaps, and $\mathbf{W_p}$ is the weight attributed to probabilities. $\sum \mid h_i - \hat{h_j} \mid$ signifies the dissimilarity between the ground truth h_i and predicted $\hat{h_j}$ heatmaps.

Following the application of the Hungarian algorithm [15], we obtain the optimal matching and categorize the predicted queries into two groups: positive queries and negative queries. Positive queries are those that align with the ground truth. For a positive query $\hat{y_i}$, we determine the matched ground truth category as g_i and the corresponding heatmap as h_i. Then, we compute the probability

loss using the Focal Loss (FL) and the heatmap loss employing the Mean Squared Error Loss (MSE). For a negative query, only the probability loss is considered. The associated ground truth g_i is designated as 0, signifying nonexistence. Thus, the overall loss can be calculated as follows:

$$\mathcal{L} = \sum_{i=1}^{M} (\lambda_p \mathrm{FL}(g_i, \hat{p_i}) + \lambda_h \mathbb{1}_{\{g_i \neq 0\}} \mathrm{MSE}(h_i, \hat{h_i})). \tag{5}$$

4 Experiments

In this section, we conduct a series of comprehensive experiments to validate the superior performance of our approach in both keypoint saliency and correspondence estimation tasks. We provide comparisons with other state-of-the-art methods to demonstrate our method's efficacy.

4.1 Implementation Details

KeypointDETR takes 3D point coordinates as input. In the point-wise encoder, we utilize 5 dynamic graph transformer modules, each with point features of 64 dimensions, the k value of k-nearest neighbors set to 20, and a global embedding dimension of 1024. In the query-wise decoder, we set the number of queries M to 50. The decoder's output comprises two branches: the heatmap head generates multi-heatmaps $\mathbf{M}_h \in \mathbb{R}^{M \times N_p}$, and the classification head provides probabilities $\mathbf{M}_p \in \mathbb{R}^{M \times C}$. For keypoint saliency estimation, C is held constant at 2. However, for keypoint correspondence estimation, C is configured as $(N_c + 1)$, where N_c signifies the number of keypoint categories. More details can be found in the *supplementary material*.

4.2 Dataset

Our experiments are conducted on the KeypointNet dataset [47], which is designed for both keypoint saliency and correspondence estimation tasks. We focus on four classical categories: airplane, chair, guitar, and table. These models are divided into training, validation, and test sets, with a distribution ratio of 7:1:2 for each category. We use 2,048 sampled points from each 3D model as input data. The results of other categories and applications of KeypointDETR can be found in the *supplementary material*.

4.3 Metrics

We evaluate our method alongside comparison methods on the keypoint saliency estimation using mIoU and Chamfer Distance (CD) while utilizing mIoU and Dual Alignment Score (DAS) [31] for keypoint correspondence estimation. The IoU is calculated as follows:

$$IoU = \frac{TP}{TP + FP + FN}, \tag{6}$$

where TP stands for true positives, FP represents false positives, and FN corresponds to false negatives. Given the ground truth and predicted keypoints sets G and K, CD can be computed as:

$$CD = \frac{1}{\mid G \mid} \sum_{p \in G} \min_{q \in K} ||p - q||_2^2 + \frac{1}{\mid K \mid} \sum_{p \in K} \min_{q \in G} ||p - q||_2^2. \tag{7}$$

The settings for clustering and NMS parameters profoundly affect the repeatability and omission rates of predicted keypoints. However, the mIoU metric used in prior methods fails to effectively capture these erroneous cases. To ensure a comprehensive comparison, we design a mIoU metric based on the Hungarian algorithm [15]. For the keypoint saliency estimation, we begin by calculating the Euclidean distance matrix between the ground truth and the predicted keypoints. Subsequently, the Hungarian algorithm is employed to determine their optimal bipartite matching. Each matched distance falling below the predefined threshold ϵ is considered a true positive. For keypoint correspondence estimation, a ground truth keypoint is considered a true positive if there is a predicted keypoint with the same semantic label whose distance to it is less than the threshold ϵ. Additionally, we further measure whether the closest keypoints between predictions and ground truth have the same semantic labels using DAS [31].

4.4 Comparison on Keypoint Saliency Estimation

We conduct a comparative analysis of our method against six advanced algorithms, including two unsupervised keypoint detection methods (Skeleton Merger [31] and UKPGAN [46]) and three point cloud geometric feature extraction algorithms (PointNet++ [28], PRA-Net [5], and DGCNN [40]), and a fully-supervised keypoint detection method (Wei *et al.* [42]). For the three point cloud geometric feature extraction algorithms on saliency analysis, the network's output is a shared heatmap, wherein each point is assigned the maximum value of the multi-heatmaps.

The quantitative results for mIoU and Chamfer Distance (CD) are presented in Tab. 1, where we compare our KeypointDETR with other algorithms using a threshold ϵ of 0.1. Fig. 6 (a) depicts the mIoU curves for various methods as ϵ varies from 0 to 0.1. It is evident that KeypointDETR consistently outperforms other methods in the Hungarian mIoU evaluation across all categories. This significant advantage stems from our method's capacity to bypass post-processing techniques like NMS or clustering algorithms, leading to direct 'one-to-one' predictions that effectively tackle redundancy and omissions in 3D keypoint detection. In contrast, methods reliant on a shared heatmap face challenges with parameter selection of post-processing, which predominantly depends on prior knowledge factors such as the expected number of keypoints and the 3D model's scale. Unfortunately, even optimal parameter settings for NMS (radius=0.1) in these methods do not entirely resolve issues of omissions and redundant predictions, as further illustrated in Fig. 4.

Table 1: Quantitative results of keypoint saliency estimation using Hungarian mIoU (%) and CD.

	Airplane		Chair		Guitar		Table		Average	
	mIoU	CD	mIoU	CD	mIoU	CD	mIoU	CD	mIoU	CD
SkeletonMerger [31]	60.73	0.107	42.74	0.189	47.71	0.132	27.84	0.302	44.75	0.182
UKPGAN [46]	62.63	0.118	25.16	0.168	54.25	0.131	17.78	0.240	39.95	0.164
PRA-Net [5]	82.88	0.049	70.44	0.060	73.71	0.053	69.69	0.053	74.18	0.053
PointNet++ [28]	82.31	0.050	57.50	0.077	72.34	0.060	60.01	0.062	68.04	0.062
DGCNN [40]	78.49	0.058	59.49	0.076	63.88	0.072	59.92	0.066	65.44	0.068
Wei *et al.* [42]	79.86	0.048	86.28	**0.055**	78.08	0.066	86.61	0.039	82.70	0.052
Ours	**91.37**	**0.045**	**91.89**	**0.055**	**93.96**	**0.045**	**98.05**	**0.027**	**93.81**	**0.043**

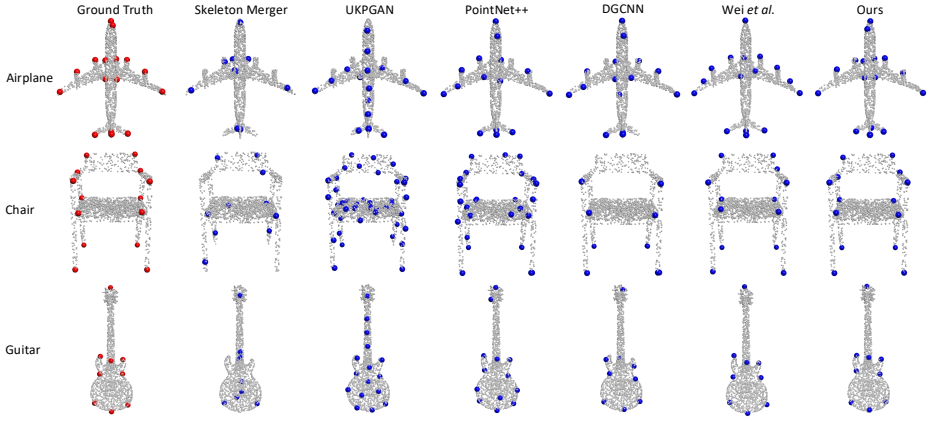


Fig. 4: Visualization examples of keypoint saliency estimation.

4.5 Comparison on Keypoint Correspondence Estimation

Keypoint correspondence estimation represents a more significant challenge compared to keypoint saliency estimation, as it not only involves locating keypoints but also predicting their semantic labels. Additionally, 3D models within the same category may not have identical semantic keypoints. Prior approaches [35, 45, 47] typically involve direct classification of points within a point cloud, which does not effectively address the issue of imbalanced positive and negative samples in keypoint detection. Some methods [42, 54] attempt to add a semantic branch to keypoint saliency estimation, aiming to assign labels to keypoints in a two-stage process.

As depicted in Tab. 2, we conduct a comparison of our method with five methods using a threshold ϵ of 0.1: PointConv [43], PointNet++ [28], DGCNN [40], PRA-Net [5], and Wei *et al.* [42]. Additionally, we present the mIoU curves for various methods, with the ϵ ranging from 0 to 0.1, in Fig. 6 (b). Fig. 5 showcases visualization results of different methods alongside the ground truth. Notably,

Table 2: Quantitative results of keypoint correspondence estimation using Hungarian mIoU (%) and DAS [31] (%).

	Airplane		Chair		Guitar		Table		Average	
	mIoU	DAS	mIoU	DAS	mIoU	DAS	mIoU	DAS	mIoU	DAS
DGCNN [40]	77.77	82.69	63.16	79.56	61.13	78.59	57.99	78.83	65.01	79.91
PointNet++ [28]	79.10	84.74	58.24	78.19	67.69	81.92	62.97	81.59	67.00	81.61
PointConv [43]	81.54	85.74	64.92	82.15	76.39	88.91	66.20	83.73	72.26	85.13
PRA-Net [5]	81.10	85.80	68.35	83.62	72.07	85.15	70.16	84.89	72.92	84.86
Wei *et al.* [42]	78.54	88.34	**79.77**	**90.44**	75.64	93.69	86.56	97.31	80.12	92.44
Ours	**85.71**	**91.74**	79.33	89.53	**90.93**	**93.89**	**92.40**	**97.79**	**87.09**	**93.23**

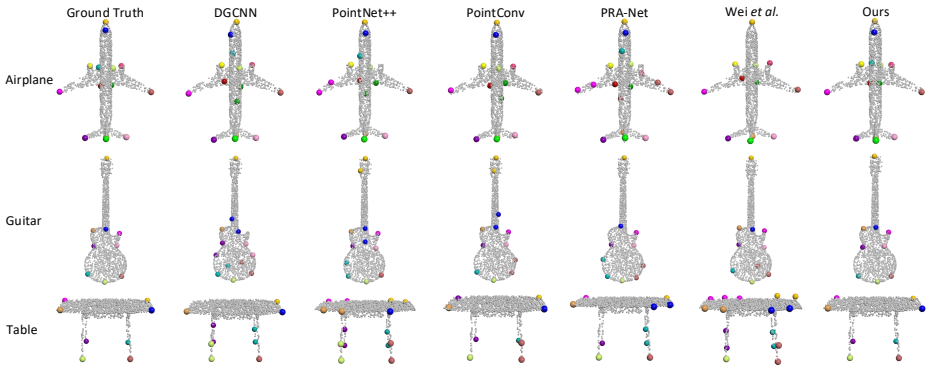


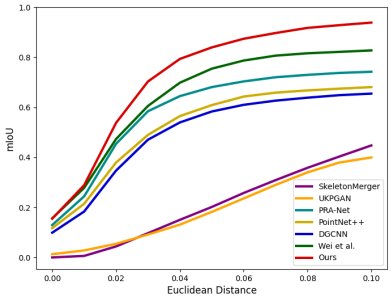
Fig. 5: Visualization examples of keypoint correspondence estimation.

our method generates no redundant points with the same label in close proximity to predicted keypoints. This demonstrates that our method adeptly accomplishes semantic correspondence for keypoints, with KeypointDETR showcasing superior performance in both the heatmap and probability branch.

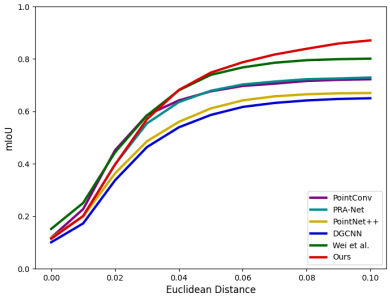
4.6 Ablation Study

We conduct a series of ablation experiments to validate the performance of KeypointDETR and deeply explore the impact of each component. Throughout these experiments, we consistently use four representative categories from the KeypointNet dataset for keypoint saliency estimation. The evaluation adopts mIoU with a threshold ϵ of 0.1.

Effects of point-wise encoder. Our point-wise encoder primarily consists of dynamic graph transformer modules that extract geometric features from the point cloud. To validate the effectiveness of our point-wise encoder, we compare KeypointDETR with its variants, which utilize different backbones for geometric feature extraction, namely DGCNN [40], PointNet++ [28], and Point Transformer [52]. The DGCNN [40] backbone (DG.Enc.) is implemented as a five-layer



(a) Hungarian mIoU curves for keypoint saliency estimation.



(b) Hungarian mIoU curves for keypoint correspondence estimation.

Fig. 6: Hungarian mIoU curves for keypoint saliency and correspondence estimation under various distance thresholds (0-0.1).

Table 3: Quantitative results of various variations in the ablation study using Hungarian mIoU (%) and CD.

	Airplane		Chair		Guitar		Table		Average	
	mIoU	CD	mIoU	CD	mIoU	CD	mIoU	CD	mIoU	CD
DG.Enc. [40]	86.75	0.048	82.12	0.062	92.76	0.052	97.62	0.037	89.81	0.049
PN2.Enc. [28]	76.30	0.071	68.31	0.101	72.09	0.099	76.42	0.099	73.28	0.092
PT.Enc. [52]	82.55	0.059	73.04	0.085	68.92	0.091	97.32	0.032	80.45	0.066
Point-BERT Enc. [49]	91.11	**0.034**	93.57	**0.034**	93.51	**0.032**	93.71	0.033	92.88	**0.033**
I2P-MAE Enc. [51]	79.06	0.050	80.40	0.067	88.79	0.048	86.05	0.048	83.57	0.053
Point-MAE Enc. [25]	89.82	0.039	**93.98**	0.055	83.79	0.043	95.95	0.043	90.88	0.045
w/o Trans.Dec.	87.96	0.044	88.23	0.058	92.74	0.045	97.77	0.029	91.67	0.044
w/o PE.	88.05	0.042	91.43	0.054	92.44	0.047	97.97	0.028	92.47	0.042
Coord-based	66.72	0.108	44.55	0.190	37.99	0.164	50.08	0.187	49.83	0.162
Coord+offsets	68.40	0.094	63.62	0.136	42.45	0.151	60.77	0.159	58.81	0.135
Ours	**91.37**	0.045	91.89	0.055	**93.96**	0.045	**98.05**	**0.027**	**93.81**	0.043

stack of EdgeConv. PointNet++ [28] (PN2.Enc.) and Point Transformer [52] (PT.Enc.) employ eight layers due to their U-net structures. Furthermore, we explore replacing our encoder with pre-trained models, including Point-BERT, Point-MAE, and I2P-MAE, and achieve favorable results. The overall quantitative comparison results are presented in the first sub-table of Tab. 3.

Effects of query-wise decoder. In the decoder phase, we develop a query-wise decoder utilizing self-attention to facilitate enhanced information exchange among queries. This decoder is tasked with decoding the heatmap and probability feature maps, resulting in the generation of multi-heatmaps and probabilities. A straightforward approach involves employing MLPs to decode these feature maps separately. However, to demonstrate the efficacy and reasoning behind our query-wise decoder design, we conduct an experiment where we replace our self-attention decoder (Trans.Dec.) with MLPs. The comparison results, as shown in the second sub-table of Tab. 3, indicate that the comprehensive performance of our KeypointDETR significantly surpasses that of its MLPs decoder variant.

Effects of position encoding in decoder. Position encoding holds a pivotal role in the self-attention mechanism and is a key factor that makes the transformer architecture suitable for 3D point cloud models, as previously validated in the Point Transformer [52]. In our investigation, we further explore the role of position encoding in our query-wise decoder, which is generated from heatmap features and point coordinates. We compare the results with and without position encoding (PE.) in the third sub-table of Tab. 3.

Coordinate-based or Heatmap-based. We further explore a coordinate-based architecture within our KeypointDETR approach. In this variant, KeypointDETR predicts the coordinates and probabilities of keypoints, and we subsequently select the closest matching points from the point cloud as the final keypoints. To enhance the accuracy of these coordinates, we incorporate a cascade module designed to predict offsets. Comparative analysis of the heatmap-based KeypointDETR and its coordinate-based variant, as shown in the fourth sub-table of Tab. 3, reveals that while the addition of offsets improves the results of the coordinate-based model, there remains a notable performance gap when compared to our primary heatmap-based KeypointDETR. This discrepancy arises because our geometric feature extraction backbone is better suited for processing 3D shapes rather than 3D spatial coordinates. Furthermore, it validates that the heatmap-based approach can achieve better performance on keypoints detection.

5 Conclusion

In this paper, we introduce KeypointDETR, an end-to-end 3D keypoint detector that obviates the need for post-processing steps, such as clustering and NMS, commonly required in heatmap-based keypoint detection methods. The training strategy of KeypointDETR involves computing a cost matrix between the predictions and ground truth, deriving an optimal bipartite matching loss. This innovative approach enables KeypointDETR to accurately predict multi-heatmaps and probabilities, which can provide the positions and semantic consistency of potential keypoints, respectively. Complementing this framework is a transformer-based network architecture. Extensive quantitative experiments and analyses substantiate the effectiveness and rationality of KeypointDETR, underlining its potential as a state-of-the-art solution in keypoint detection.

Acknowledgements

This work is partially supported by the following grants: National Natural Science Foundation of China (No. 62172363), Management Center of the School of Software (Ningbo) at Zhejiang University under Grant (No. Z24001).

References

1. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.L.: D3feat: Joint learning of dense detection and description of 3d local features. In: CVPR. pp. 6359–6367 (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020)
3. Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3d mesh saliency with statistical descriptors. In: Comput. Graph. Forum. vol. 27, pp. 643–652. Wiley Online Library (2008)
4. Chen, R., Ma, Y., Chen, N., Liu, L., Cui, Z., Lin, Y., Wang, W.: Structure-aware long short-term memory network for 3d cephalometric landmark detection. IEEE Transactions on Medical Imaging **41**(7), 1791–1801 (2022)
5. Cheng, S., Chen, X., He, X., Liu, Z., Bai, X.: Pra-net: Point relation-aware network for 3d point cloud analysis. IEEE TIP **30**, 4436–4448 (2021)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fernandez-Labrador, C., Chhatkuli, A., Paudel, D.P., Guerrero, J.J., Demonceaux, C., Gool, L.V.: Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In: ECCV. pp. 546–563. Springer (2020)
8. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**, 187–199 (2021)
9. Hampali, S., Sarkar, S.D., Rad, M., Lepetit, V.: Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In: CVPR. pp. 11090–11100 (2022)
10. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: CVPR. pp. 11632–11641 (2020)
11. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3d segmentation of point clouds. In: CVPR. pp. 2626–2635 (2018)
12. Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., Kanazawa, A.: Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In: CVPR. pp. 12783–12792 (2021)
13. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. The Vis. Comput. **21**, 649–658 (2005)
14. Khoury, M., Zhou, Q.Y., Koltun, V.: Learning compact geometric features. In: ICCV. pp. 153–161 (2017)
15. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
16. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: ACM SIGGRAPH 2005 Papers, pp. 659–666 (2005)
17. Li, H., Guo, Z., Rhee, S.M., Han, S., Han, J.J.: Towards accurate facial landmark detection via cascaded transformers. In: CVPR. pp. 4176–4185 (2022)
18. Li, J., Lee, G.H.: Usip: Unsupervised stable interest point detection from 3d point clouds. In: ICCV. pp. 361–370 (2019)
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)

20. Liu, Z., Wu, Z., Tóth, R.: Smoke: Single-stage monocular 3d object detection via keypoint estimation. In: CVPR. pp. 996–997 (2020)
21. Mian, A.S., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE TPAMI **28**(10), 1584–1601 (2006)
22. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: ICPR. vol. 3, pp. 850–855. IEEE (2006)
23. Novatnack, J., Nishino, K.: Scale-dependent 3d geometric features. In: ICCV. pp. 1–8. IEEE (2007)
24. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3d object detection with pointformer. In: CVPR. pp. 7463–7472 (2021)
25. Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: ECCV. pp. 604–621. Springer (2022)
26. Park, C., Jeong, Y., Cho, M., Park, J.: Fast point transformer. In: CVPR. pp. 16949–16958 (2022)
27. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)
28. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS **30** (2017)
29. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. science **344**(6191), 1492–1496 (2014)
30. Salti, S., Tombari, F., Spezialetti, R., Di Stefano, L.: Learning a descriptor-specific 3d keypoint detector. In: ICCV. pp. 2318–2326 (2015)
31. Shi, R., Xue, Z., You, Y., Lu, C.: Skeleton merger: an unsupervised aligned keypoint detector. In: CVPR. pp. 43–52 (2021)
32. Shu, Z., Yu, J., Chao, K., Xin, S., Liu, L.: A multi-modal attention-based approach for points of interest detection on 3d shapes. IEEE TVCG (2024)
33. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. The Vis. Comput. **27**, 963–976 (2011)
34. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Comput. Graph. Forum. vol. 28, pp. 1383–1392. Wiley Online Library (2009)
35. Sung, M., Su, H., Yu, R., Guibas, L.J.: Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions. NeurIPS **31** (2018)
36. Tang, J., Gong, Z., Yi, R., Xie, Y., Ma, L.: Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In: CVPR. pp. 1726–1735 (2022)
37. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: ECCV. pp. 356–369. Springer (2010)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
39. Wang, H., Guo, J., Yan, D.M., Quan, W., Zhang, X.: Learning 3d keypoint descriptors for non-rigid shape matching. In: ECCV. pp. 3–19 (2018)
40. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM TOG **38**(5), 1–12 (2019)
41. Wei, G., Cui, Z., Zhu, J., Yang, L., Zhou, Y., Singh, P., Gu, M., Wang, W.: Dense representative tooth landmark/axis detection network on 3d model. Computer Aided Geometric Design **94**, 102077 (2022)
42. Wei, G., Ma, L., Wang, C., Desrosiers, C., Zhou, Y.: Multi-task joint learning of 3d keypoint saliency and correspondence estimation. Computer-Aided Design **141**, 103105 (2021)

43. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR. pp. 9621–9630 (2019)
44. Yew, Z.J., Lee, G.H.: 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In: ECCV. pp. 607–623 (2018)
45. Yi, L., Su, H., Guo, X., Guibas, L.J.: Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In: CVPR. pp. 2282–2290 (2017)
46. You, Y., Liu, W., Ze, Y., Li, Y.L., Wang, W., Lu, C.: Ukpgan: A general self-supervised keypoint detector. In: CVPR. pp. 17042–17051 (2022)
47. You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., Wang, W.: Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In: CVPR. pp. 13647–13656 (2020)
48. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: ICCV. pp. 12498–12507 (2021)
49. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: CVPR. pp. 19313–19322 (2022)
50. Yuan, H., Zhao, C., Fan, S., Jiang, J., Yang, J.: Unsupervised learning of 3d semantic keypoints with mutual reconstruction. In: ECCV. pp. 534–549. Springer (2022)
51. Zhang, R., Wang, L., Qiao, Y., Gao, P., Li, H.: Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In: CVPR. pp. 21769–21780 (2023)
52. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV. pp. 16259–16268 (2021)
53. Zhong, C., You, P., Chen, X., Zhao, H., Sun, F., Zhou, G., Mu, X., Gan, C., Huang, W.: Snake: Shape-aware neural 3d keypoint field. NeurIPS **35**, 7052–7064 (2022)
54. Zhu, X., Du, D., Huang, H., Ma, C., Han, X.: 3d keypoint estimation using implicit representation learning. arXiv preprint arXiv:2306.11529 (2023)
55. 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)