OSP2B: One-Stage Point-to-Box Network for 3D Siamese Tracking

Jiahao Nie[1] , **Zhiwei He**[1*] , **Yuxiang Yang**[1] , **Zhengyi Bao**[1] , **Mingyu Gao**[1] , **Jing Zhang**[2]
[1]School of Electronics and Information, Hangzhou Dianzi University, China
[2]School of Computer Science, The University of Sydney, Australia
{jhnie, zwhe, yyx, baozhengyi, mackgao}@hdu.edu.cn, jing.zhang1@sydney.edu.au

Abstract

Two-stage point-to-box network acts as a critical role in the recent popular 3D Siamese tracking paradigm, which first generates proposals and then predicts corresponding proposal-wise scores. However, such a network suffers from tedious hyper-parameter tuning and task misalignment, limiting the tracking performance. Towards these concerns, we propose a simple yet effective one-stage point-to-box network for point cloud-based 3D single object tracking. It synchronizes 3D proposal generation and center-ness score prediction by a parallel predictor without tedious hyper-parameters. To guide a task-aligned score ranking of proposals, a center-aware focal loss is proposed to supervise the training of the center-ness branch, which enhances the network's discriminative ability to distinguish proposals of different quality. Besides, we design a binary target classifier to identify target-relevant points. By integrating the derived classification scores with the center-ness scores, the resulting network can effectively suppress interference proposals and further mitigate task misalignment. Finally, we present a novel one-stage Siamese tracker OSP2B equipped with the designed network. Extensive experiments on challenging benchmarks including KITTI and Waymo SOT Dataset show that our OSP2B achieves leading performance with a considerable real-time speed.

1 Introduction

Single object tracking (SOT) is a fundamental task in computer vision and contributes to various applications, such as autonomous driving and mobile robotics [Zhang and Tao, 2020; Javed *et al.*, 2022]. Early tracking methods mainly focus on the 2D image domain. With the developments of LiDAR sensors, and considering that 3D point cloud data captured by LiDAR is more robust to adverse weather and illumination than RGB data, increasing efforts [Giancola *et al.*, 2019; Fang *et al.*, 2020; Qi *et al.*, 2020; Zhou *et al.*, 2022; Zheng *et al.*, 2022] are devoted to point cloud-based tracking.

*Corresponding author

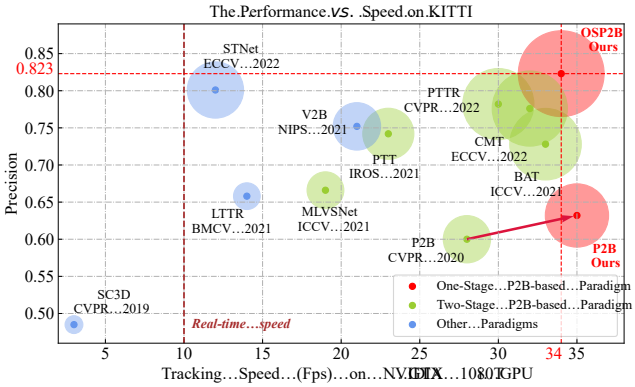

Figure 1: Comparison with SOTA methods on KITTI. We report the Precision performance with respect to tracking speed on a single NVIDIA GTX 1080Ti GPU. Circle size indicates the overall performance regarding Precision and tracking speed.

In this paper, we study the problem of 3D SOT on LiDAR point clouds.

Recently, the Siamese network-based tracking paradigm has attracted remarkable attention. Current mainstream Siamese trackers such as P2B [Qi *et al.*, 2020], BAT [Zheng *et al.*, 2021] and PTTR [Zhou *et al.*, 2022] rely on a two-stage point-to-box network to *learn the offsets of seed points towards object's center to generate 3D proposals (first stage), and then predict the highest scoring one as tracking box (second stage)*. Despite the great success, the two-stage design suffers from some inherent shortcomings: 1) A series of predefined hyper-parameters that require empirical and heuristic configurations. For example, tracking performance is sensitive to the number of proposals as reported in [Qi *et al.*, 2020]. 2) To predict proposal-wise scores, such a two-stage network defines proposals within a threshold distance from the object's center in 3D Euclidean space as positive samples for training. However, the nearest proposal has an equal contribution to the training as other positive ones, which exposes a task misalignment problem, i.e., the predicted highest-scoring proposal is not guaranteed to be the most accurate one.

Towards these concerns, we aim to offer a one-stage solution for 3D Siamese tracking. Inspired by one-stage 2D Siamese trackers [Guo *et al.*, 2020; Li *et al.*, 2018], the key to the one-stage design is to perform the proposal generation

and proposal-wise score prediction simultaneously. Differently, in 2D tracking, the proposal-wise scores are predicted directly from image pixels without location information of the proposals, while for 3D point cloud tracking, seed points covering the surface of an object need to be offset towards the object's center to generate proposals for further score reasoning [Qi *et al.*, 2019]. Here, we argue that the seed point features that generate proposals are embedded with rich geometric cues, therefore it is feasible to synchronously predict the proposal-wise scores from these features. Additionally, we discover that sufficiently accurate proposals can usually be generated by offset learning (as verified in Section 4.2), so how to predict the most accurate one as a tracking result (i.e., task alignment) is significant for tracking task. To guide a task-aligned score ranking of proposals, distinguishing positive samples of different quality for training matters. Moreover, false-positive samples also need to be paid attention to avoid affecting the score ranking.

Motivated by the above analysis, we propose a simple yet effective **one-stage point-to-box network** that is free of tedious hyper-parameters, to improve the 3D Siamese tracking paradigm. Specifically, we design a parallel predictor and use the same seed point features to synchronize proposals generation and their center-ness scores prediction. Due to the 3D rigid object's constant size (width, height, and length) in point cloud sequences, the center-ness scores can effectively represent the accuracy of proposals. To distinguish positive samples of different quality to solve the task misalignment problem, we propose a center-aware focal loss to train the center-ness branch, in which a center-aware mask is devised to assign different loss weights for samples with regard to their proximity to the object's center. In addition, we develop a target classifier to classify foreground target points and background interference points. Leveraging classification scores, the center-ness scores can be refined to suppress false positive proposals to further alleviate the task misalignment. Finally, by integrating the proposed one-stage point-to-box network as the prediction head, a novel one-stage Siamese tracking method dubbed **OSP2B** is presented. As shown in Fig. 1, OSP2B achieves state-of-the-art (SOTA) tracking performance, while running at a high speed of 34 frames per second (Fps). In particular, our one-stage point-to-box network outperforms the previous two-stage counterpart in both accuracy and efficiency, as clearly verified by P2B $v.s.$ P2B-ours in Fig. 1.

The main contributions of this paper are as follows:

- We propose the first one-stage point-to-box network to improve the 3D Siamese tracking paradigm, and present a novel OSP2B tracker to deal with point cloud-based single object tracking.
- We design a parallel predictor to synchronize 3D proposal generation and center-ness score prediction, avoiding tracking-sensitive hyper-parameters.
- We design a center-aware focal loss and a target classifier to guide a task-aligned score ranking of proposals, effectively addressing the task misalignment problem.
- Compared with SOTA methods, our OSP2B outperforms them in terms of both accuracy and efficiency on challenging benchmarks including KITTI and Waymo SOT Dataset.

2 Related Work

2.1 2D Siamese Tracking

Currently, Siamese network-based tracking methods [Bertinetto *et al.*, 2016; Li *et al.*, 2018; Guo *et al.*, 2020; Chen *et al.*, 2021] serve a dominant role in 2D visual object tracking. Generally, the Siamese tracking paradigm composed of two branches projects the target template and search images into an intermediate feature embedding space, and then fuses the template and search features by fusion modules such as cross-correlation [Zhang *et al.*, 2020; Nie *et al.*, 2022a] and attention-based operators [Nie *et al.*, 2022c; Pi *et al.*, 2022]. Subsequently, the fused features are further used to regress bounding boxes and calculate box-wise scores. Despite of the great success, it is non-trivial to extend 2D Siamese techniques to process 3D point cloud data.

2.2 3D Siamese Tracking

Early 3D tracking methods [Asvadi *et al.*, 2016; Liu *et al.*, 2018; Pieropan *et al.*, 2015; Bibi *et al.*, 2016; Kart *et al.*, 2019] directly employ the 2D Siamese architecture to process RGB-D data with an additional depth channel. Recently, many efforts have been focused on tracking point cloud objects, as point cloud data is less sensitive to adverse weather than RGB-D data. As a pioneer, SC3D [Giancola *et al.*, 2019] proposes the first 3D Siamese tracker, but it is not an end-to-end framework and fails to run in real-time due to exhaustive 3D candidate boxes. To address these issues, P2B [Qi *et al.*, 2020] introduces a two-stage point-to-box network to perform proposal generation and proposal-wise score prediction for tracking, making a good balance between accuracy and speed. Inspired by this strong baseline, a series of follow-ups have been presented. BAT [Zheng *et al.*, 2021] and PTTR [Zhou *et al.*, 2022] improve P2B by using different feature fusion modules to replace the point-wise correlation operator. PTT [Shan *et al.*, 2021], MLVSNet [Wang *et al.*, 2021] and GLT-T [Nie *et al.*, 2022b] propose to form a powerful feature presentation of seed points by designing more advanced structures. Although great progress has been made, these methods all follow the two-stage point-to-box network-based paradigm. By contrast, we offer a simpler and more effective one-stage design for 3D Siamese tracking.

2.3 Point-to-Box Network

Point-to-box network is inspired by Hough voting [Qi *et al.*, 2019], where a set of seed points are sampled to generate votes from their features, and the votes are targeted to reach the object's center. To apply the idea of Hough voting to 3D single object tracking, existing two-stage point-to-box network-based trackers first generate votes (i.e., proposals) by offsetting the seed points covering the surface of an object to its center, and form vote clusters via a shared PointNet [Qi *et al.*, 2017a], including set-abstraction and propagation layers. With the vote clusters, a genetic point set learning network is then employed to predict scores and orientations of the votes, and refine the votes through secondary offset

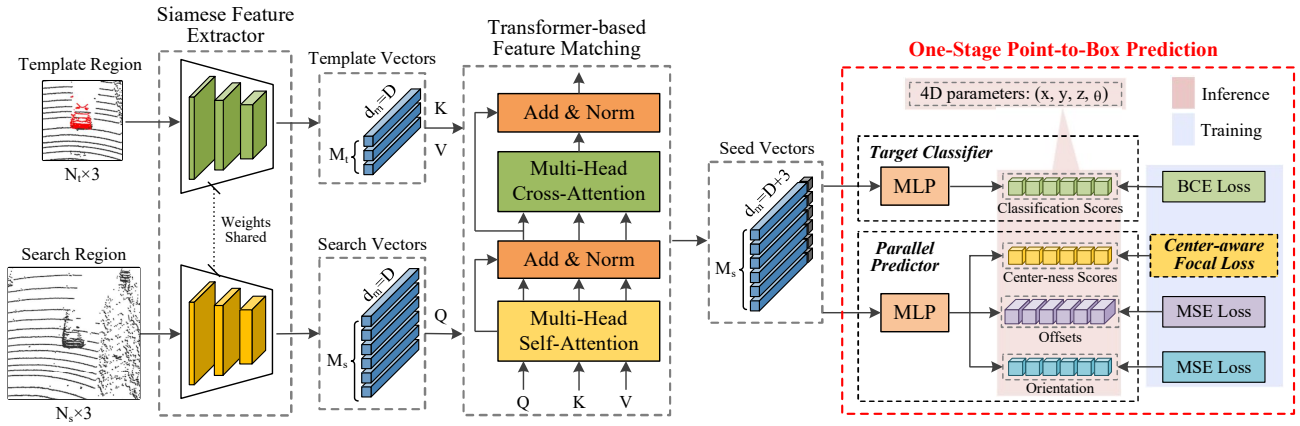

Figure 2: Overview of the proposed **OSP2B**. Given a template and search region, we first utilize a Siamese Feature Extractor to extract the point features, and fuse them with a Transformer-based feature matching module to output seed points. Finally, we apply the proposed One-Stage Point-to-Box Prediction network to predict a final 3D bounding box (BBox).

learning. In contrast, within the proposed one-stage point-to-box network, we synchronize proposal generation by a parallel predictor, and customize a center-aware focal loss and a target classifier to address the task misalignment problem.

3 OSP2B: A Novel One-Stage Point-to-Box Network based 3D Siamese Tracker

3.1 Overall Architecture

In a 3D scene, given a template target $\mathbf{P}^t = \{p_i^t\}_{i=1}^{N_t}$ cropped by the 3D bounding box (BBox) in the first frame, the tracking task aims to locate this target in search region $\mathbf{P}^s = \{p_i^t\}_{i=1}^{N_s}$ frame by frame. The 3D BBox is parameterized with a 7-dimensional vector, where (x, y, z) and (w, h, l) represent the center coordinate and size, while θ is the orientation. Since the object size is known in the first frame and keeps constant, only (x, y, z, θ) need to be predicted for tracking. To this end, we propose OSP2B, a novel one-stage paradigm for point cloud object tracing. As shown in Fig. 2, our OSP2B consists of three key parts: Siamese feature extractor, transformer-based feature matching, and one-stage point-to-box prediction.

Siamese Feature Extractor. Following previous trackers [Qi *et al.*, 2020; Zheng *et al.*, 2021], we employ a modified PointNet++ [Qi *et al.*, 2017b] as the backbone to subsample key points and extract their semantic features. More concretely, we remove the last task-relevant layers of PointNet++ and use it to encode multi-scale point features of template and search.

Transformer-based Feature Matching. Transformer-based feature matching is adopted to fuse the point features of the template and search to generate seed points. Similar to existing methods [Zhou *et al.*, 2022; Chen *et al.*, 2021], we first exploit a shared self-attention block [Dosovitskiy *et al.*, 2020; Xu *et al.*, 2022; Zhang *et al.*, 2022a; Zhang *et al.*, 2022b] to enhance the feature representation of template and search, and then a cross-attention block is used to match them.

One-Stage Point-to-Box Prediction. Different from the existing two-stage point-to-box network (The detailed structure comparison is offered in the supplementary material), we propose a one-stage point-to-box network as prediction head to predict 4-dimensional BBox parameters (x, y, z, θ), as introduced in Section 3.2. We first design a parallel predictor to generate proposals and predict center-ness scores simultaneously. Then, a center-aware focal loss is devised to train the center-ness branch to distinguish positive samples with different quality. Finally, we also develop a target classifier to identify foreground target points and background interference points to suppress false positive samples.

3.2 One-Stage Point-to-Box Network

Parallel Predictor. Given the seed point vectors $\mathbf{V} = \{v_i \in [f_i; p_i]\}_{i=1}^{M_s}$ as inputs, where f_i and $p_i = (x_i, y_i, z_i)$ denote the semantic features and 3-dimensional coordinate, we design an offset branch and a center-ness branch to form a parallel predictor, which generates proposals and predicts center-ness scores, respectively. Meanwhile, an orientation branch is attached to predict the orientation of each proposal, as shown in the red box in Fig. 2.

For proposal generation, the offset branch outputs a set of 3-dimensional vectors $\{d_i = (x_i^d, y_i^d, z_i^d)\}_{i=1}^{M_s}$ to predict the distances from the seed points $\{p_i\}_{i=1}^{M_s}$ to object's center. The proposals $\{p_i + d_i\}_{i=1}^{M_s}$ are obtained by applying offsets to the original coordinates of seed points. During training, let $(\tilde{x}, \tilde{y}, \tilde{z})$ represent the center of ground truth BBox, the target offset for the i-th seed point can be calculated by:

$$t_i[0] = \tilde{x} - x_i, \quad t_i[1] = \tilde{y} - y_i, \quad t_i[2] = \tilde{z} - z_i. \tag{1}$$

With $\{t_i\}_{i=1}^{M_s}$, we sample foreground target points and use MSE loss function to compute the offset loss as:

$$\mathcal{L}_{off} = \frac{1}{M_s'} \sum_{i=1}^{M_s'} \sum_{j=0}^{2} \|d_i[j] - t_i[j]\|^2, \tag{2}$$

where $M_s' < M_s$ is the number of foreground target points.

For center-ness score prediction, the goal of the center-ness branch is to output scores that can denote the accuracy of proposals. To this end, a common solution typically calculates

the distance between proposals and the object's center in 3D Euclidean space and collects the seed points corresponding to the proposals within a threshold as positive samples while treating other points as negative ones. However, this approach ignores the spatial distribution of proposals inside objects, especially slender objects, resulting in unbalanced positive and negative samples. We thereby propose shape-adaptive labels to balance the positive and negative sample sizes. In practice, instead of a sphere, the positive sample candidates are specified as in a rectangular cube that occupies by the object at scale τ:

$$\tilde{s}_i = \begin{cases} 1 & \text{if } (x_i + x_i^d, y_i + y_i^d, z_i + z_i^d) \in \mathbb{R}^{\tau(\tilde{w},\tilde{h},\tilde{l})} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\tilde{s}_i$ is the label for seed point p_i, $\tilde{w}$, $\tilde{h}$ and $\tilde{l}$ denotes the width, height, and length of ground truth BBox, respectively. The center-ness loss is computed by a center-aware focal loss function, which is detailed below.

To characterize the orientations of cubic objects in 3D space, we train the orientation branch by:

$$\mathcal{L}_{ori} = \frac{1}{M_s'} \sum_{i=1}^{M_s'} \|\theta_i - \tilde{\theta}\|^2, \tag{4}$$

where θ_i and $\tilde{\theta}$ denote the predicted orientation of each proposal and the orientation of ground truth BBox.

Center-aware Focal Loss. In fact, the center-ness scores should relate to the proximity of proposals to the object's center. Therefore, a training strategy that can distinguish positive samples of different proximity is desired to train the center-ness branch. To achieve this, we customize a center-aware focal loss. Specifically, a center-aware point mask is defined as:

$$mask_i = \sqrt[3]{\frac{\min(l,r)}{\max(l,r)} \times \frac{\min(t,b)}{\max(t,b)} \times \frac{\min(f,k)}{\max(f,k)}}, \tag{5}$$

where l, r, t, b, f and k represent the distance of proposal point $p_i + d_i$ to the left, right, top, bottom, front and back surfaces of ground truth BBox, respectively. In this way, the samples closer to the object's center tend to have higher mask scores. Here, we consider point mask scores as loss weights for different positive samples, thereby incorporating center-aware geometry prior into the model training. To this end, the center-aware focal loss is formulated as:

$$\mathrm{CAFL}(s_i, \tilde{s}_i) = \begin{cases} -\alpha(1-s_i)^\gamma (1 + mask_i)\log(s_i) & \tilde{s}_i = 1 \\ -\beta(s_i)^\gamma \log(1-s_i) & \tilde{s}_i = 0, \end{cases} \tag{6}$$

where s_i is the predicted center-ness score of i-th proposal. Following [Lin *et al.*, 2017], we empirically set the three hyper-parameters α=2, β=1, and γ=2 to balance the loss weights for positive (easy) and negative (hard) samples. The center-ness loss is:

$$\mathcal{L}_{cen} = \frac{1}{M_s} \sum_{i=1}^{M_s} \mathrm{CAFL}(s_i, \tilde{s}_i). \tag{7}$$

Target Classifier. Since background interference points are not supervised in offset training, these points might be sampled as positive samples for center-ness training, resulting in proposals far from the center being predicted high scores. To make the tracker resist interference, a target classifier is also devised. We take the foreground seed points inside the 3D object BBox as positive samples and the other points as negative ones. Using the vanilla binary cross-entropy loss function, the classifier loss can be defined as:

$$\mathcal{L}_{cla} = -\frac{1}{M_s} \sum_{i=1}^{M_s} (\tilde{c}_i \log(c_i) + (1-\tilde{c}_i)\log(c_i)), \tag{8}$$

where c_i is the predicted classification score, $\tilde{c}_i$ =1 or 0 is the corresponding label.

Considering the different contributions of seed points to offset learning, we also use classification scores to supervise the offset branch training except for adjusting center-ness scores. The offset loss in Eq. 2 is further refined as:

$$\mathcal{L}_{off} = \frac{1}{M_s'} \sum_{i=1}^{M_s'} (\sum_{j=0}^{2} \|d_i[j] - t_i[j]\|^2)(1 + c_i), \tag{9}$$

where c_i allows the model to focus more on the points inside the object, facilitating offset learning and consequently generating better proposals.

3.3 Implementation

Model Inputs. During training, we sample paired samples, i.e., template and search regions from the same point cloud sequences. The template region is formed by merging the points inside the ground truth BBoxes of $(t-1)$-th frame and 1-st frame. For the search region, since the target shifts a little between consecutive frames, only a prior region where the target may appear is required. We thereby enlarge the ground truth BBox of t-th frame by 2 meters and crop the points within this enlarged area. To enhance the robustness of the model, we randomly impose small shifts to the BBoxes along the x, y, and z axes in the training phase.

During inference, the ground truth BBox of 1-st frame is given, but the ground truth BBox of subsequent frames is unknown. Therefore, when generating the template region and search region of t-th frame, the ground truth BBox of $(t-1)$-th and t-th frames mentioned above is replaced by the BBox of $(t-1)$-th frame predicted by the model.

Model Details. We randomly sample $N_t = 512$ and $N_s = 1024$ points for the template region and search region, respectively. Then a modified PointNet++ [Qi *et al.*, 2017b] with 3 set-abstraction layers is adopted as the Siamese backbone, to obtain the semantic features of key points, where $M_t = 64$, $M_s = 128$ and $D = 256$. In the proposed one-stage point-to-box prediction head, the hidden layers are built by a 2-layer MLP that has a constant channel dimension.

Training. Our OSP2B can be trained in an end-to-end manner. With the above losses of four branches in the one-stage point-to-box network, the total loss is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{off} + \lambda_2 \mathcal{L}_{ori} + \lambda_3 \mathcal{L}_{cen} + \lambda_4 \mathcal{L}_{cla}, \tag{10}$$

where λ_1, λ_2, λ_3 and λ_4 are all set to 1 to balance these losses. We use Adam optimizer to train the OSP2B model

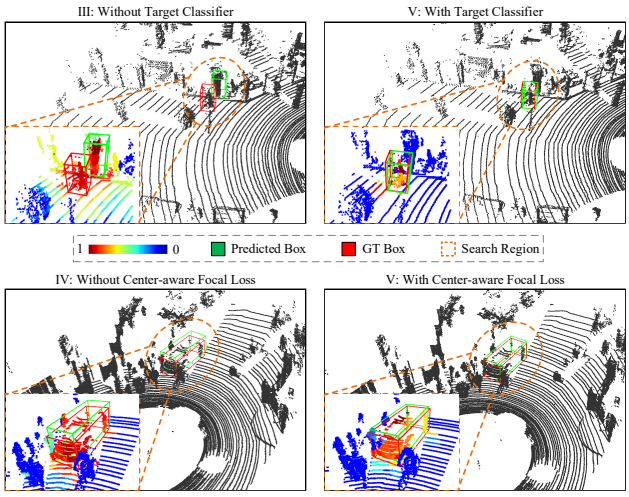

Figure 3: Visualization of tracking results without or with target classifier (top row) and center-aware focal loss (bottom row), respectively. The final scores of proposals range from 0 to 1.

on 4 NVIDIA GTX 1080Ti GPUs for 160 epochs. The initial learning rate is set to 0.001 and decreased by a linear decay factor of 0.2 every 40 epochs. We provide the implementation code in the supplement.

Inference. To predict 4-dimensional vector (x, y, z, θ) of the target object frame by frame, OSP2B produces a 6-dimensional vector $(x_i^d, y_i^d, z_i^d, \theta_i, s_i, c_i)$ for each seed point $p_i = (x_i, y_i, z_i)$. The inference phase can be formulated as:

$$I = \underset{i}{\operatorname{argmax}}\{s_i \times c_i\}, \tag{11}$$

where I is the index of the best proposal predicted by the OSP2B model, and the 4-dimensional vector (x, y, z, θ) of final tracking BBox is calculated by:

$$x = x_I + x_I^d, \quad y = y_I + y_I^d, \quad z = z_I + z_I^d, \quad \theta = \theta_I. \tag{12}$$

4 Experiments

4.1 Experimental Settings

Datasets. To evaluate our model, we conduct comprehensive experiments, including the ablation study on KITTI (Section 4.2) and comparison with SOTA methods on KITTI and Waymo SOT Dataset (Section 4.3). KITTI [Geiger *et al.*, 2012] contains 21 training LiDAR sequences and 29 test LiDAR sequences. Due to the labels of test data is not open, we split the training set for training, validation, and testing, following the previous works [Qi *et al.*, 2020]. Waymo SOT Dataset [Zhou *et al.*, 2022] is a more challenging and large-scale dataset, recently collected from the raw Waymo data [Sun *et al.*, 2020]. To be a fair comparison, we perform training and testing based on the method described in [Zhou *et al.*, 2022].

Evaluation Metrics. Following the common practice, we calculate *Success* and *Precision* metrics by One Pass Evaluation (OPE) [Wu *et al.*, 2013] to report tracking performance. *Success* measures the intersection over union (IOU) between predicted BBox and ground truth BBox. *Precision* measures the distance between the centers of two BBoxes.

	Parallel Predictor	Center-aware Focal Loss	Target Classifier	*Success*	*Precision*
I				64.3	76.4
II	✓			$65.5_{\uparrow 1.2}$	$78.3_{\uparrow 1.9}$
III	✓	✓		$66.8_{\uparrow 2.5}$	$80.7_{\uparrow 4.3}$
IV	✓		✓	$66.1_{\uparrow 1.8}$	$79.6_{\uparrow 3.2}$
V	✓	✓	✓	$\mathbf{67.5}_{\uparrow 3.2}$	$\mathbf{82.3}_{\uparrow 5.9}$

Table 1: Ablation study of model components on Car category from KITTI. **Bold** denotes the best result.

4.2 Ablation Study

In this section, we conduct a series of ablation studies, including qualitative and quantitative analyses to validate the effectiveness of the proposed one-stage point-to-box network. As previous works [Giancola *et al.*, 2019; Qi *et al.*, 2020; Zheng *et al.*, 2021], all ablated experiments are carried out on the Car category from the KITTI dataset.

Model Component. To investigate the contributions of the designed components: parallel predictor, center-aware focal loss, and target classifier to the tracking performance, a component-wise ablation experiment is conducted. We report the results of OSP2B using different prediction heads, as shown in Table 1. Compared to the two-stage point-to-box network (I), the proposed one-stage point-to-box network with only parallel predictor (II) outperforms it by 1.2% and 1.9% in terms of Success and Precision, respectively, proving the effectiveness of our one-stage design in synchronizing 3D proposals generation and center-ness scores prediction. For (III), the center-aware focal loss distinguishes positive samples with different quality in training, which increases the probability of the most accurate proposal being selected, and thus a significant performance improvement of 2.5% and 4.3% in Success and Precision is achieved. For (IV), the target classifier suppresses the interference points and guides the network to predict more accurate scores, also leading to better performance. In addition, we visualize the tracking results of (III $v.s.$ V) and (IV $v.s.$ V) in Fig. 3 to intuitively demonstrate the effectiveness of the center-aware focal loss and the target classifier. As can be seen, the background interference points are given low scores by using the target classifier (top row), and the points closer to the object's center have higher scores than those far away by using the center-aware focal loss (bottom row). When combining all components (V), we achieve the best Success and Precision of 67.5% and 82.3%.

Effectiveness of Task Alignment. Here, we present the effectiveness of task alignment by visualizing IoU $v.s.$ Score of 3D proposals. From Fig. 4, we have two observations: 1) The overall accuracy of the proposals generated by our proposed one-stage prediction head is slightly lower than that of the two-stage head, but the accuracy of the best one is comparable; 2) In our one-stage head, the better proposals are more likely to be predicted as the tracking results. In other words, the predicted scores are better aligned with the quality of the proposals. The two observations imply that predicting the best proposal as the tracking box contributes more remarkably to the tracker, compared to refining the proposal using the two-stage design. Owing to the proposed center-ness fo-

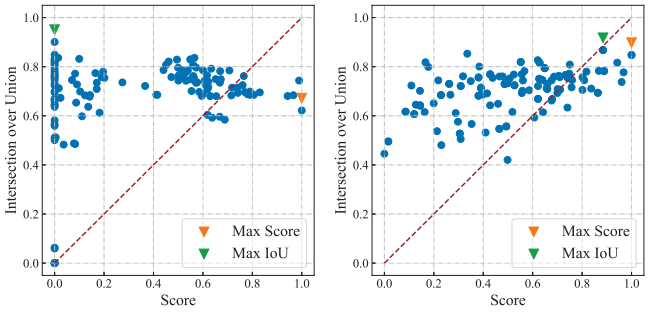

Figure 4: Visualization of the IoU $v.s.$ Score. Left: two-stage point-to-box network. Right: one-stage point-to-box network (ours).

Scale Rate τ	*Success*	*Precision*
0.3	64.7	76.6
0.4	67.3	81.9
0.5	**67.5**	**82.3**
0.6	66.9	80.4
0.7	65.7	78.6

Table 2: Influence of different scale rate τ on Car category from KITTI. **Bold** denotes the best result.

cal loss and target classifier, the task misalignment problem is effectively alleviated in our one-stage prediction head.

Scale Rate. Scale rate τ is a significant hyper-parameter in our proposed one-stage point-to-box network. Too small a value will result in insufficient positive samples for training, while too large a value will distract the model and affect its discriminative ability. Therefore, we conduct an experiment to determine the optimal value of τ. As reported in Table 2, when $\tau = 0.5$, the best *Success* and *Precision* values are obtained. So we set τ to 0.5 for all experiments if not specified.

4.3 Comparison with SOTA Methods

Results on KITTI. We compare our OSP2B with 9 SOTA methods on four categories from KITTI [Geiger *et al.*, 2012]. The results are presented in Table 3. OSP2B achieves state-of-the-art performance in all categories. Especially for Pedestrian, an impressive performance advantage over other comparison methods is exhibited. Besides, we obtain the best mean Precision of 82.3%, which suggests that our method is able to accurately predict object centers. Compared to the most recent method STNet [Hui *et al.*, 2022], the proposed method not only shows competitive performance in Car, Van, and Cyclist categories but also surpasses it in the Pedestrian category by a large margin. Moreover, OSP2B runs nearly three times faster than STNet on the same experimental setting (34 Fps $v.s.$ 12 Fps). Notably, PTTR [Zhou *et al.*, 2022] has a similar architecture to our OSP2B, except for the prediction head. Compared to it, OSP2B shows significant performance improvements in all categories, such as the Car category (*Success*: 65.2%→67.5%; *Precision*: 77.4%→82.3%), which manifests the superiority of our one-stage design. In addition, to intuitively compare the proposed one-stage point-to-box network and the two-stage point-to-box network used in PTTR, we also visualize their results from the four categories. As shown in Fig. 5, our method can track different categories of point cloud objects more accurately and robustly.

	Category Frame Number	Car 6,424	Ped 6,088	Van 1,248	Cyc 308	Mean 14,068
Success	SC3D [Giancola *et al.*, 2019]	41.3	18.2	40.4	41.5	31.2
	P2B [Qi *et al.*, 2020]	56.2	28.7	40.8	32.1	42.4
	MLVSNet [Wang *et al.*, 2021]	56.0	34.1	52.0	34.3	45.7
	LTTR [Cui *et al.*, 2021]	65.0	33.2	35.8	66.2	48.7
	BAT [Zheng *et al.*, 2021]	60.5	42.1	52.4	33.7	51.2
	PTT [Shan *et al.*, 2021]	67.8	44.9	43.6	37.2	55.1
	V2B [Hui *et al.*, 2021]	70.5	48.3	50.1	40.8	58.4
	PTTR [Zhou *et al.*, 2022]	65.2	50.9	52.5	65.1	57.9
	CMT [Guo *et al.*, 2022]	70.5	49.1	54.1	55.1	59.4
	STNet [Hui *et al.*, 2022]	**72.1**	49.9	**58.0**	**73.5**	**61.3**
	OSP2B (ours)	67.5	**53.6**	56.3	65.6	60.5
Precision	SC3D [Giancola *et al.*, 2019]	57.9	37.8	47.0	70.4	48.5
	P2B [Qi *et al.*, 2020]	72.8	49.6	48.4	44.7	60.0
	MLVSNet [Wang *et al.*, 2021]	74.0	61.1	61.4	44.5	66.6
	LTTR [Cui *et al.*, 2021]	77.1	56.8	48.4	89.9	65.8
	BAT [Zheng *et al.*, 2021]	77.7	70.1	67.0	45.4	72.8
	PTT [Shan *et al.*, 2021]	81.8	72.0	52.5	47.3	74.2
	V2B [Hui *et al.*, 2021]	81.3	73.5	58.0	49.7	75.2
	PTTR [Zhou *et al.*, 2022]	77.4	81.6	61.8	90.5	78.2
	CMT [Guo *et al.*, 2022]	81.9	75.5	64.1	82.4	77.6
	STNet [Hui *et al.*, 2022]	**84.0**	77.2	**70.6**	**93.7**	80.1
	OSP2B (ours)	82.3	**85.1**	66.2	90.5	**82.3**

Table 3: Comparison on Car, Pedestrian, Van, and Cyclist categories from KITTI benchmark. **Bold** and underline denote the best result and the second-best one, respectively.

	Category Frame Number	Veh 53,377	Ped 27,308	Cyc 5,374	Mean 86,095
Success	SC3D [Giancola *et al.*, 2019]	46.5	26.4	26.5	33.1
	P2B [Qi *et al.*, 2020]	55.7	35.3	30.7	40.6
	PTTR [Zhou *et al.*, 2022]	58.7	**49.0**	**43.3**	**50.3**
	OSP2B (ours)	**59.2**	46.6	43.0	49.6
Precision	SC3D [Giancola *et al.*, 2019]	52.7	37.8	37.6	42.7
	P2B [Qi *et al.*, 2020]	62.2	54.9	44.5	53.9
	PTTR [Zhou *et al.*, 2022]	65.2	**69.1**	60.4	64.9
	OSP2B (ours)	**67.3**	67.4	**62.5**	**65.7**

Table 4: Comparison on Vehicle, Pedestrian, and Cyclist categories from Waymo SOT Dataset benchmark. **Bold** and underline denote the best result and the second-best one, respectively.

Results on Waymo SOT Dataset. To further evaluate the proposed OSP2B method, we also conduct comparison experiments on the large-scale dataset Waymo SOT Dataset [Zhou *et al.*, 2022]. We select SC3D [Giancola *et al.*, 2019], P2B [Qi *et al.*, 2020] and PTTR [Zhou *et al.*, 2022], which have reported performance on this dataset as comparison methods. As presented in Table 4, OSP2B achieves state-of-the-art performance in all categories, demonstrating that our one-stage method not only performs well on the small-scale dataset but also delivers satisfactory results on the large-scale dataset. Besides, since the Waymo SOT Dataset benchmark contains a wide range of complex real-world scenes, the superior performance of the proposed method indicates that it has great potential for practical applications.

Inference Speed. In addition to tracking accuracy comparisons, we also compare the inference speed of our OSP2B

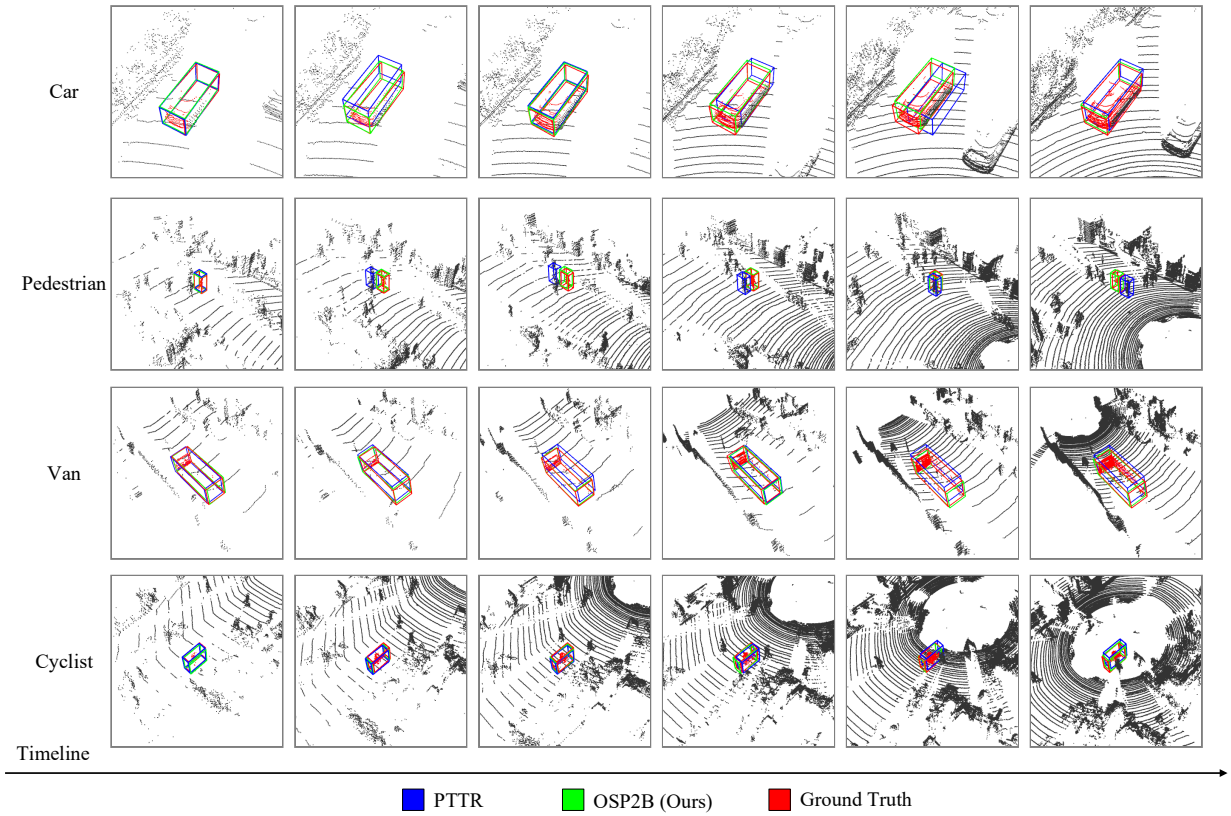

Figure 5: Visual tracking results of our OSP2B and PTTR on the point cloud sequences of Car, Pedestrian, Van, and Cyclist categories. The foreground points within the ground truth BBoxes are colored in red.

Method	Fps	Method	Fps
SC3D [Giancola *et al.*, 2019]	3	PTT [Shan *et al.*, 2021]	23
P2B [Qi *et al.*, 2020]	28	V2B [Hui *et al.*, 2021]	21
MLVSNet [Wang *et al.*, 2021]	19	PTTR [Zhou *et al.*, 2022]	30
LTTR [Cui *et al.*, 2021]	14	STNet [Hui *et al.*, 2022]	12
BAT [Zheng *et al.*, 2021]	33	OSP2B (ours)	**34**

Table 5: Speed comparison on all test frames in the Car category from KITTI. **Bold** denotes the best result.

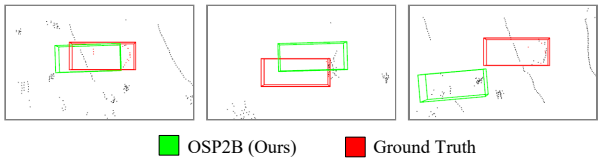

Figure 6: Tracking failure cases of our OSP2B on extremely sparse point cloud scene.

with SOTA methods. For a fair comparison, the average running time of each tracker is calculated on all test frames in the Car category from KITTI. OSP2B runs at 34 Fps on a single NVIDIA 1080Ti GPU, including 7.6 ms for processing point cloud, 21.1 ms for network forward propagation, and 0.8 ms for post-processing. The running speeds of other methods under the same workstation are reported in Table 5. Thanks to the efficient one-stage point-to-box prediction head, our OSP2B achieves the fastest inference speed.

5 Limitation Discussion

We show the tracking failure cases of our OSP2B in Fig. 6. It can be seen that OSP2B is not ready to handle extremely sparse point cloud scenes. This is mainly owing to the inability of the model to infer offsets from a small number of points, and thus causing 3D proposals to drift from the object center. One possible solution is to use the point cloud completion technique to obtain informative model inputs.

6 Conclusion

In this paper, we revisit the 3D single object tracking on LiDAR point clouds, and propose to boost the Siamese paradigm with a novel one-stage point-to-box network, which is demonstrated to be superior over the two-stage counterpart by comprehensive experiments and analysis. By integrating this network as a prediction head, we develop a one-stage Siamese tracking method OSP2B, which can track point cloud objects in a one-stage manner and effectively address the task misalignment problem. Benefiting from the one-stage design, our OSP2B significantly outperforms previous SOTA trackers in terms of both accuracy and efficiency on challenging datasets. We hope OSP2B could serve as a one-stage baseline method and inspire future research on accurate and efficient 3D single object trackers.

Acknowledgements

This work was supported by the Zhejiang Provincial Natural Science Foundation Key Fund of China (LZ23F030003), the Fundamental Research Funds for the Provincial Universities of Zhejiang (GK239909299001-003), the Zhejiang Provincial Major Research and Development Project of China (2023C01132), the Hangzhou Major Science and Technology Innovation Project of China (2022AIZD0009), and the Zhejiang Provincial Key Lab of Equipment Electronics.

References

[Asvadi *et al.*, 2016] Alireza Asvadi, Pedro Girao, Paulo Peixoto, and Urbano Nunes. 3d object tracking using rgb and lidar data. In *ITSC*, pages 1255–1260, 2016.

[Bertinetto *et al.*, 2016] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[Bibi *et al.*, 2016] Adel Bibi, Tianzhu Zhang, and Bernard Ghanem. 3d part-based sparse tracker with automatic synchronization and registration. In *CVPR*, pages 1439–1448, 2016.

[Chen *et al.*, 2021] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021.

[Cui *et al.*, 2021] Yubo Cui, Zheng Fang, Jiayao Shan, Zuoxu Gu, and Sifan Zhou. 3d object tracking with transformer. In *BMCV*, 2021.

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

[Fang *et al.*, 2020] Zhang Fang, Suo Zhou, Yu Cui, and Sac Scherer. 3d-siamrpn: An end-to-end learning method for real-time 3d single object tracking using raw point cloud. *IEEE Sensors Journal*, 21(4):1019–1026, 2020.

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[Giancola *et al.*, 2019] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging Shape Completion for 3D Siamese Tracking. In *CVPR*, pages 1359–1368, 2019.

[Guo *et al.*, 2020] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, pages 6269–6277, 2020.

[Guo *et al.*, 2022] Zhiyang Guo, Yunyao Mao, Wengang Zhou, Min Wang, and Houqiang Li. Cmt: Context-matching-guided transformer for 3d tracking in point clouds. In *European Conference on Computer Vision*, pages 95–111, 2022.

[Hui *et al.*, 2021] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3D Siamese Voxel-to-BEV Tracker for Sparse Point Clouds. In *NIPS*, pages 28714–28727, 2021.

[Hui *et al.*, 2022] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3D Siamese Transformer Network for Single Object Tracking on Point Clouds. In *ECCV*, 2022.

[Javed *et al.*, 2022] Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, Michael Felsberg, and Jiri Matas. Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

[Kart *et al.*, 2019] Ugur Kart, Alan Lukezic, Matej Kristan, Joni-Kristian Kamarainen, and Jiri Matas. Object tracking by reconstruction with view-specific discriminative correlation filters. In *ICCV*, pages 1339–1348, 2019.

[Li *et al.*, 2018] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017.

[Liu *et al.*, 2018] Ye Liu, Xiao-Yuan Jing, Jianhui Nie, Hao Gao, Jun Liu, and Guo-Ping Jiang. Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in rgb-d videos. *TMM*, 21(3):664–677, 2018.

[Nie *et al.*, 2022a] Jiahao Nie, Zhiwei He, Yuxiang Yang, Mingyu Gao, and Zhekang Dong. Learning localization-aware target confidence for siamese visual tracking. *TMM*, 2022.

[Nie *et al.*, 2022b] Jiahao Nie, Zhiwei He, Yuxiang Yang, Mingyu Gao, and Jing Zhang. Glt-t: Global-local transformer voting for 3d single object tracking in point clouds, 2022.

[Nie *et al.*, 2022c] Jiahao Nie, Han Wu, Zhiwei He, Mingyu Gao, and Zhekang Dong. Spreading fine-grained prior knowledge for accurate tracking. *TCSVT*, 2022.

[Pi *et al.*, 2022] Zhixiong Pi, Weitao Wan, Chong Sun, Changxin Gao, Nong Sang, and Chen Li. Hierarchical feature embedding for visual tracking. In *European Conference on Computer Vision*, pages 428–445, 2022.

[Pieropan *et al.*, 2015] Alessandro Pieropan, Niklas Bergström, Masatoshi Ishikawa, and Hedvig Kjellström. Robust 3d tracking of unknown objects. In *ICRA*, pages 2410–2417, 2015.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 30, 2017.

[Qi *et al.*, 2019] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019.

[Qi *et al.*, 2020] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2B: Point-to-Box Network for 3D Object Tracking in Point Clouds. In *CVPR*, pages 6328–6337, 2020.

[Shan *et al.*, 2021] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. PTT: Point-Track-Transformer Module for 3D Single Object Tracking in Point Clouds. In *IROS*, pages 1310–1316, 2021.

[Sun *et al.*, 2020] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 3d siamese transformer network for single object tracking on point clouds. In *CVPR*, pages 2446–2454, 2020.

[Wang *et al.*, 2021] Zhoutao Wang, Qian Xie, Yu-Kun Lai, Jing Wu, Kun Long, and Jun Wang. MLVSNet: Multi-level Voting Siamese Network for 3D Visual Tracking. In *ICCV*, pages 3081–3090, 2021.

[Wu *et al.*, 2013] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418, 2013.

[Xu *et al.*, 2022] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022.

[Zhang and Tao, 2020] Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IoT*, 8(10):7789–7817, 2020.

[Zhang *et al.*, 2020] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *European Conference on Computer Vision*, pages 771–787, 2020.

[Zhang *et al.*, 2022a] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022.

[Zhang *et al.*, 2022b] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: Learning varied-size window attention in vision transformers. In *European conference on computer vision*, 2022.

[Zheng *et al.*, 2021] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-Aware Feature Enhancement for Single Object Tracking on Point Clouds. In *ICCV*, pages 13179–13188, 2021.

[Zheng *et al.*, 2022] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2022.

[Zhou *et al.*, 2022] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. PTTR: Relational 3D Point Cloud Object Tracking with Transformer. In *CVPR*, pages 8531–8540, 2022.