

3D Single Object Tracking with Multi-View Unsupervised Center Uncertainty Learning

Chengpeng Zhong¹, Hui Shuai¹, Jiaqing Fan², Kaihua Zhang¹, and Qingshan Liu¹ ✉

ABSTRACT

Center point localization is a major factor affecting the performance of 3D single object tracking. Point clouds themselves are a set of discrete points on the local surface of an object, and there is also a lot of noise in the labeling. Therefore, directly regressing the center coordinates is not very reasonable. Existing methods usually use volumetric-based, point-based, and view-based methods, with a relatively single modality. In addition, the sampling strategies commonly used usually result in the loss of object information, and holistic and detailed information is beneficial for object localization. To address these challenges, we propose a novel Multi-view unsupervised center Uncertainty 3D single object Tracker (MUT). MUT models the potential uncertainty of center coordinates localization using an unsupervised manner, allowing the model to learn the true distribution. By projecting point clouds, MUT can obtain multi-view depth map features, realize efficient knowledge transfer from 2D to 3D, and provide another modality information for the tracker. We also propose a former attraction probability sampling strategy that preserves object information. By using both holistic and detailed descriptors of point clouds, the tracker can have a more comprehensive understanding of the tracking environment. Experimental results show that the proposed MUT network outperforms the baseline models on the KITTI dataset by 0.8% and 0.6% in precision and success rate, respectively, and on the NuScenes dataset by 1.4%, and 6.1% in precision and success rate, respectively. The code is made available at <https://github.com/abcchears/MUT.git>.

KEYWORDS

3D single object tracking; uncertainty modeling; multi-view feature; holistic and detailed descriptor

The task of 3D Single Object Tracking (3D SOT) involves estimating and tracking the 3D position and orientation of a specific object in a sequence of point clouds, with the initial 3D Bounding Box (3D BBox) serving as a reference. This problem has considerable importance in the domain of autonomous driving^[1-3], where 3D SOT can assist robots in perceiving their surroundings, identifying and tracking objects of interest, and achieving safe and intelligent navigation and control. In addition, 3D SOT has significant implications in the field of robot vision^[4,5], where it can enable robots to learn and interact with their surroundings. By accurately tracking the position and orientation of objects of interest, robots can more effectively navigate and manipulate their environment, opening up a broader range of possibilities in manufacturing, healthcare, and home automation. As shown in Fig. 1, the mainstream 3D SOT methods mainly follow the P2B paradigm^[7] or the motion-centric paradigm^[6]. The P2B paradigm first establishes the association between the template and search area and then uses an independent detection module^[8] to locate the object. However, the P2B paradigm only considers the appearance information of point clouds, leading to significant performance loss when the object undergoes deformation.

M2Track^[6] proposes a two-stage motion-centric tracking network that extracts motion information by calculating the relative displacement of the bounding box. This enables the tracker to perform well even when the object shape changes drastically. M2Track^[6] uses a segmentation network to obtain the approximate position of the object, and then uses the detection

module to return a finer bounding box. Regardless of whether it is the P2B paradigm or the motion-centric paradigm, object detection relies on independent detectors. Some studies^[9, 10] have pointed out that the incorrect localization of the center point coordinates is the main factor leading to poor 3D detection performance. The point cloud is a discrete set of points on the object's surface, which is non-structural and discrete. It is challenging for the detector to directly predict the object structure using detectors. Besides, the point cloud is difficult to annotate manually, leading to inevitable noise in the label for model training. Therefore, directly regressing the center coordinates is not reasonable. Existing methods usually adopt volumetric-based, point-based, and view-based methods. However, these methods use relatively single modalities, while different modalities can complement each other. The mainstream method for extracting detailed descriptors is to use PointNet++^[11], but the downsampling strategy leads to the loss of many object points. Furthermore, both holistic and detailed information is useful for object localization. Holistic descriptors provide a complete understanding of the point cloud scene, while detailed descriptors provide specific details of the object. In recent years, the 3D SOT field has made significant progress, but most models somehow ignore the aforementioned issues, which can lead to parameter drift of the tracker.

Therefore, we propose a method called Multi-view unsupervised center Uncertainty 3D single object Tracker (MUT). Since the uncertainty is not annotated, MUT uses an unsupervised strategy to learn the distribution of the 3D BBox center

1 School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China

2 School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Address correspondence to Qingshan Liu, qslu@nuist.edu.cn

© The author(s) 2023. The articles published in this open access journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

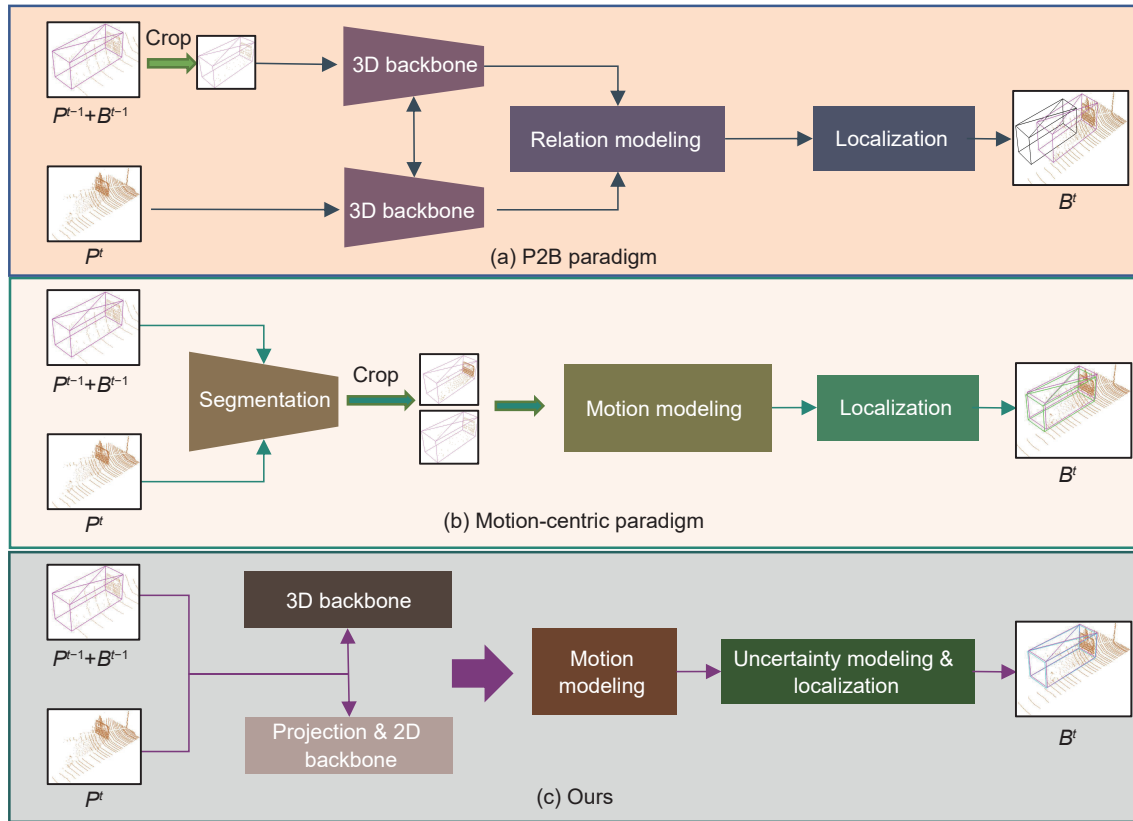


Fig. 1 Comparison of different paradigms in 3D SOT field. P2B paradigm uses predicted 3D BBox to crop the point cloud of the previous frame, obtaining the template. Then it establishes the association between the template and the search area and uses an independent detection module to locate the object. M2Track^[4] proposes a two-stage motion-centric tracking network. It first uses a segmentation network to obtain the rough position of the object, and then extracts motion information by calculating the relative displacement of the bounding box. Then it uses a detection module to return more refined bounding boxes. However, both of them lack attention to the object center coordinates, so MUT combines 3D point cloud and 2D depth map modalities, and optimizes tracker prediction by modeling the uncertainty of object center coordinates. Red indicates ground truth, black indicates P2B^[7] predicted BBox, green indicates M2Track^[4] predicted BBox, and blue indicates MUT predicted box.

coordinates, making the predicted distribution close to the true distribution. It is worth noting that we only use unsupervised learning for uncertainty, but our network is still supervised. In the training process, MUT focuses on optimizing the predictions with higher uncertainty and can obtain satisfactory 3D bounding boxes even in extreme cases.

MUT also projects the point cloud onto a 2D plane, generating a multi-view depth map that preserves the original attributes of the point cloud. The depth map enables the model to capture the shape information of objects. By leveraging advanced 2D encoders, we can not only effectively achieve knowledge transfer from 2D to 3D, but also combine information from both modalities. Additionally, we propose to enhance feature learning by using the point cloud's Holistic and Detailed Descriptors (HDDs). Through the use of the former attraction probability sampling strategy, we can preserve object information. The holistic descriptors contain complete information about the scene, while the detailed descriptors provide specific details. By combining the two, the tracker can gain a more accurate understanding of the tracking scene. We integrate these modules into a unified network framework and achieve excellent performance on the KITTI dataset^[12] and NuScenes dataset^[13]. Compared to the baseline model, the precision and success rate on the KITTI dataset are improved by 0.8% and 0.6%, respectively, and on the NuScenes dataset^[13] are improved by 1.4% and 6.1%, respectively. The contributions of this paper can be summarized as follows:

- We model the uncertainty in object localization using an unsupervised manner, which allows the model to learn the true distribution of the center coordinates and optimize the poorly performing predicted 3D BBoxes.
- We capture multi-view depth features to provide another modality information while enabling efficient 2D-to-3D knowledge transfer.
- We propose sampling strategies that more effectively preserve object information while exploiting both holistic and detailed descriptors, allowing trackers to gain a more comprehensive understanding of the point cloud scene.
- By integrating the aforementioned components into a unified network framework, we obtain a more powerful tracker, called MUT. Our approach achieves satisfactory performance on the KITTI and NuScenes datasets, outperforming baseline models in both precision and success rate.

1 Related Work

1.1 3D single object tracking

Early 3D SOT models predominantly relied on RGB-D (“D” represents depth map information) data as input. However, with the rapid development of 3D acquisition devices, a new trend in SOT methods^[7, 14, 15] emerged, which builds models based on 3D LiDAR point clouds. The first exploration in this direction was initiated by SC3D^[14], which proposed a 3D Siamese tracking method based on shape completion that can extract geometric

features from sparse point clouds and regularize the encoding with an auto-encoder network to generate latent representations with semantic geometric information. SC3D^[14] also suggested that 3D object tracking and 3D shape completion are complementary tasks and that learning a more meaningful latent representation can improve tracking performance. However, the use of a simple convolutional encoder and a fully connected decoder in SC3D^[14] may not capture the complex geometric structures and details of 3D objects.

To overcome this limitation, SiamRPN^[16], a Siamese network-based tracker, was proposed with a Region Proposal Network (RPN) to perform bounding box regression and classification on the cross-correlation features between the template and the search area. To speed up computation, SiamRPN^[16] proposes a depth-wise cross-correlation layer and a multi-task loss function to jointly optimize the classification and regression branches. However, it suffers from the lack of strict translation invariance, which means that the cross-correlation features may not be consistent under different translations of the object, and its ability to perceive 3D object relationships is limited by the use of a bird's eye view. To address these limitations, P2B^[7] extends SiamRPN^[16] to 3D space and uses votenet^[8] to detect the object information after incorporating it into the search region. However, the approach of P2B^[7] greatly reduces the number of seed points, so MLVNet^[17] retains the seed point information through a multi-layer hough voting network. Furthermore, M2Track^[6] proposes a two-stage motion-centric paradigm to track the object by learning the motion information of consecutive frames, as the Siamese paradigm-based tracking network ignores the motion information between consecutive frames. However, most of the aforementioned methods to some extent ignore the high uncertainty of the center coordinate localization. To address this issue, an unsupervised approach is employed to model the uncertainty of the 3D BBox center and obtain a more interpretable tracking network.

1.2 Uncertainty in deep learning

Deep learning typically models uncertainty using Bayesian theory. In Bayesian modeling^[18], two main types of uncertainty are typically studied: epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty describes the uncertainty in model parameters, while aleatoric uncertainty captures the uncertainty in observed noise. Gaussian YOLOv3^[19] explores the application of uncertainty in object detection by modeling the uncertainty of predicted 2D boxes to calibrate detection scores. Monodle^[9] analyzes the fitting performance of various subtasks in monocular 3D object detection and finds that the main factor affecting detection performance is the localization error of the center coordinate. Therefore, Monodle^[9] proposes incorporating heteroscedastic aleatoric uncertainty modeled by Laplace likelihood into localization estimation. Meanwhile, MonoFlex^[20] combines the idea of predicting the height ratio of object depth with uncertainty theory to build an ensemble learning method to estimate object center positions. MonoRun^[21] uses an uncertainty-based region reconstruction network to regress pixel-related 2D-3D coordinates and learn dense associations and object geometry in a self-supervised manner. GLENet^[10] models one-to-many relations between typical 3D objects and themselves using a conditional variational autoencoder to construct a probabilistic detector and supervise the learning of localization uncertainty. We incorporate heteroscedastic uncertainty with Gaussian distribution into the center coordinate localization and learn the true distribution using an unsupervised manner, thus improving the model's generalization.

2 Method

2.1 Tracking process

The framework of the MUT network is illustrated in Fig. 2. Given the continuous frame point clouds P^t and P^{t-1} and the predicted BBox B^{t-1} at time $t-1$, the Multi-view Visual Feature Extractor

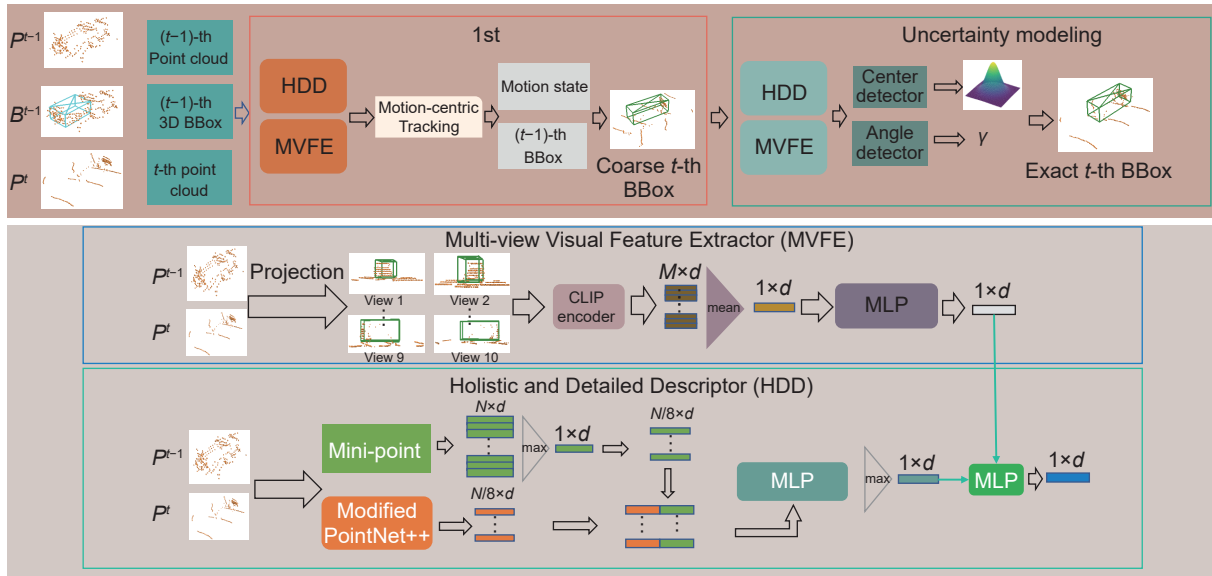


Fig. 2 Overall architecture of MUT. Given two consecutive point clouds and the predicted BBox of the previous frame, in the 1st stage, MUT projects the point clouds from different perspectives onto a 2D plane and extracts depth map features using a pre-trained CLIP^[22]. At the same time, the HDD module is used to obtain relevant descriptive information about the point cloud. Next, MUT uses the motion-centric tracking module to obtain the 3D BBox of the previous frame and the relative displacement between the two frames, and combines them to get a coarse 3D BBox. Finally, to obtain a more accurate 3D BBox, MUT uses the center detector to predict the Gaussian distribution of the object centroid, uses the angle detector to predict the offset angle γ , and then combines them for object localization.

(MVFE) projects the point cloud into a 2D space from multiple perspectives, and obtains depth maps that retain the characteristics of the original point cloud. The pre-trained CLIP^[22] is then used to extract semantic features from the depth map. Meanwhile, the HDD is used to extract the features of the point cloud. The holistic and detailed descriptors are combined with the depth features and fed into the motion-centric tracking module^[6] to obtain the 3D BBox of the previous frame and the relative displacement between the two frames. These are then combined to generate a rough 3D BBox. As obtaining an accurate 3D BBox is challenging, the uncertainty modeling module in MUT is employed to refine the bounding box. To this end, MUT first decouples the object bounding box parameters (x, y, z) and γ belonging to different solution spaces. Then, the center detector and angle detector are used to predict the centroid distribution and offset angle γ , respectively.

2.2 Problem statement

The objective of the 3D single object tracking task is to locate an object in subsequent frames, given its bounding box in the first frame. The point cloud at time t is denoted as $P^t = \{p_i^t | i = 1, 2, \dots, N\}$, where p_i^t represents a vector that represents the 3D coordinates (x_i^t, y_i^t, z_i^t) , and N is the number of point clouds. The initial bounding box in the first frame is defined as $B = (x, y, z, w, l, h, \gamma)$, where (x, y, z) represents the center coordinates, and (w, l, h) represents the object's width, length, and height, while γ denotes the angle of rotation of the object around the y -axis.

It is assumed for this task that the size of the rigid object remains constant across frames, while for non-rigid objects, the corresponding bounding box size is represented by its maximum value in the scene. We utilize the point cloud from the $t-1$ frame along with the bounding box, and the point cloud from the t frame as inputs. The goal is to output the object's position in the t frame. The process can be described in the following:

$$\text{Tracker: } \mathbb{R}^{N \times 3} \times \mathbb{R}^{N \times 3} \times \mathbb{R}^7 \rightarrow \mathbb{R}^4 \quad (1)$$

$$\text{Tracker: } (P^{t-1}, P^t, B^{t-1}) \mapsto (x, y, z, \gamma) \quad (2)$$

2.3 MVFE

Most of the current work only uses the information of a single modality for tracking. Multimodal information is more beneficial for tracking. Therefore, MUT proposes MVFE, which can obtain depth maps from different perspectives, thereby obtaining additional 2D knowledge supplements. As shown in Fig. 2, given the consecutive point clouds P^{t-1} and P^t and the predicted bounding box B^{t-1} at time $t-1$, MVFE selects ten viewing angles and then projects the point clouds onto the 2D plane to obtain the corresponding projection images. Then, the pre-trained CLIP^[22] is applied to extract deep features. While the default model ViT^[23] in CLIP^[22] is powerful, we choose to use the Resnet50^[24] provided by CLIP^[22] to balance efficiency and performance. To fully utilize the information from different viewpoints, mean pooling is used to obtain fused depth image features. Then, the fused depth image features are input to mini-PointNet^[25] to further refine the features to obtain F^m .

2.4 HDD

MUT uses the HDD module to extract and utilize both holistic and detailed descriptors of the point cloud. Specifically, given two consecutive frames P^{t-1} and P^t and an object bounding box B^{t-1} as

inputs, we first use mini-PointNet^[25] to obtain the holistic descriptor of the point cloud. Meanwhile, we use the modified PointNet++^[11] to obtain the detailed descriptor of the point cloud. The farthest point sampling strategy used by PointNet++^[11] can lead to a reduction in the number of object points and has a high computational cost. Therefore, we propose the former attraction probability sampling strategy. This sampling strategy is based on the probability that each point belongs to the object point, and the points with higher probability are preferentially selected. We feed P^{t-1} and P^t into an MLP to obtain the probability P_o of each point belonging to the object point.

$$P_o = \text{MLP}(P^{t-1}, P^t) \quad (3)$$

MLP consists of nine fully connected layers, batchnorm layers and ReLU activation functions (except for the last layer). The channel numbers of each MLP layer are set to [64, 64, 64, 128, 1024, 512, 256, 128, 128]. The subsequent PointNet++^[11] will be downsampled based on the probability vector P_o . Once the detailed descriptor F^d and the holistic descriptor F^h of the point cloud are obtained, they are combined and fed into the MLP for fusion. The maximum value of the fused features is then taken to obtain the geometric features F^p of the point cloud.

$$F^p = \max(\text{MLP}_{\text{concat}}(F^d, F^h)) \quad (4)$$

The obtained point cloud features F^p and depth map features F^m are spliced and input to the MLP network for fusion, and finally the point cloud features F^n with geometric and semantic information are obtained.

$$F^n = \text{MLP}(F^m, F^p) \quad (5)$$

2.5 Estimation of uncertainty

From Fig. 3, it can be observed that the reduction of object points leads to increased difficulty in predicting the center point, which means that the uncertainty of the center point prediction is influenced by the number of object points. Once the object points decrease to a certain degree, the model's output may be indistinguishable from "random guessing". To alleviate this situation, we adopt the maximum likelihood explanation of the Gaussian error model, enabling the model to learn the distribution in the data using an unsupervised manner and reducing the interference of data ambiguity. Building on previous work^[10], we assume that the distribution of centroid coordinates follows a Gaussian distribution $\mathbb{N}(\mu, \sigma^2)$ and aim to model $P(B^t | P^t)$, where μ represents the mean of the Gaussian distribution and σ represents the standard deviation. Here, B^t and P^t represent the 3D bounding box and point cloud at time t , respectively. We treat the variance as the model's uncertainty and use the optimization process described in Ref. [26] to maximize the logarithmic likelihood function.

$$P(B^t | P^t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[B^t - Y]^2}{2\sigma^2}\right) \quad (6)$$

where Y denotes the manually labeled bounding box centroid. In addition, the model uses the angle detector to predict the angle γ of the object bounding box rotation around the up-axis.

Center detector. The structure of the center detector is shown in Fig. 4. The center detector consists of three MLP layers, each layer consisting of a linear layer, batch normalization layer, and GELU activation function except for the last layer. Additionally, an additional activation function is applied to σ to ensure it is

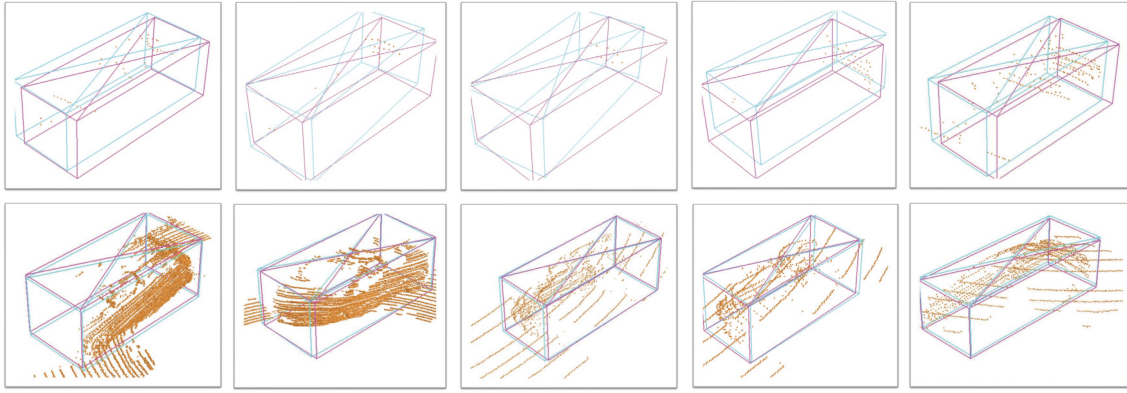


Fig. 3 Main challenge of our model is the uncertainty in the localization of object centers. Red represents the ground truth, while blue indicates the predicted bounding boxes. It can be observed that as the object points become sparser, the predicted center location may deviate from the ground truth and the uncertainty in the prediction increases. This could potentially lead to bias in the learned model parameters.

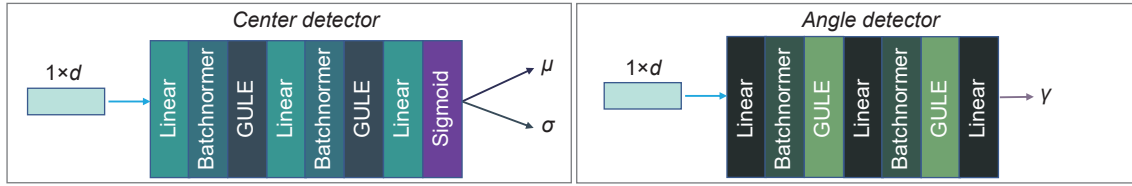


Fig. 4 Structure of center detector and angle detector.

greater than or equal to 0.

Angle detector. The structure of the angle detector is also shown in Fig. 4. Although the composition of the angle detector is similar to that of the center detector, the number of channels in the linear layer is four times that of the center detector, which allows the angle detector to better regress γ .

2.6 Loss function

Consistent with previous work^[6], in the first stage, we use the standard cross-entropy loss to obtain the foreground point mask, motion information between adjacent frames, and rough predicted bounding boxes, denoted as $\mathcal{L}_{\text{segmentation}}$, $\mathcal{L}_{\text{motion}}$, and $\mathcal{L}_{\text{bbox_1st}}$, respectively. In the second stage, we model the uncertainty by minimizing the negative log-likelihood loss function $\mathcal{L}_{\text{bbox_2nd}}$, thus encouraging the model to optimize the point cloud bounding box with high uncertainty variance.

$$\mathcal{L}_{\text{bbox_2nd}} = \frac{1}{2} [\log(\max(\sigma^2, \epsilon)) + \frac{(Y - \mu)^2}{\max(\sigma^2, \epsilon)}] \quad (7)$$

where μ represents the mean value of the centroid distribution, σ represents the standard deviation of the distribution, Y represents the ground truth, and ϵ is a constant used to prevent the network from failing to converge when σ approaches 0. Meanwhile, we use the smooth L1 loss^[27] to regress the angle γ , denoted as $\mathcal{L}_{\text{angle}}$. The final loss function can be expressed as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{segmentation}} + \lambda_2 \mathcal{L}_{\text{motion}} + \lambda_3 \mathcal{L}_{\text{bbox_1st}} + \lambda_4 \mathcal{L}_{\text{bbox_2nd}} + \lambda_5 \mathcal{L}_{\text{angle}} \quad (8)$$

After grid search, we set the parameters as follows: $\lambda_1 = 0.1$, $\lambda_2 = 2$, $\lambda_3 = 2$, $\lambda_4 = 2$, and $\lambda_5 = 10$.

3 Experiment

3.1 Experiment setups

Input. To form P^{t-1} and P^t , we sample 1024 points from frames $t-1$ and t , respectively. Moreover, we randomly horizontally flip

two object points and their corresponding bounding box by rotating them around their up-axis within the range of $[-10^\circ, 10^\circ]$, which is consistent with the approach used in Ref. [6].

Network architecture. We employ the PointNet^[25] and the modified PointNet++^[11] as our backbone. The PointNet^[25] consists of 9 MLP layers, excluding the last layer, and each layer utilizes LayerNorm layer and LeakyReLU activation function. To replace the farthest point sampling strategy in the original PointNet++^[11], we use the former attraction probability sampling strategy. Additionally, we trim the PointNet++^[11] to use three Set Abstraction (SA) layers with radius designed to be 0.3, 0.5, and 0.7 m. The output feature dimension is set to $d = 256$.

Datasets. We evaluate our proposed method on two benchmark datasets commonly used in the SOT task: the KITTI dataset^[12] and the NuScenes dataset^[13]. The KITTI dataset^[12] consists of 21 outdoor scenes with 8 different types of objects, while the more challenging NuScenes dataset^[13] contains 1000 driving scenes for 23 object classes. To maintain consistency with previous works^[7, 14, 28], we divide the KITTI dataset^[12] into training, validation, and test sets. However, as the real labels of the test set are not publicly available, we rely on the provided splits to ensure fairness in comparison. In contrast, we use the training and validation sets of the NuScenes dataset^[13] to train and test our model.

Training and testing details. The network is trained using the Adam optimizer^[29] with a batch size of 48 for 60 stages. The initial learning rate is set to 0.001, which is reduced by a factor of 10 after every 20 stages. For testing, we apply the trained model to track objects in subsequent frames of the test sequence using the MUT approach on a frame-by-frame basis.

3.2 Evaluation metrics

For the evaluation metrics, we follow the baseline model^[6] using OPE^[30] to measure success rate and precision. We report success rate and precision for each model in the following experiments. Given a network output of the tracking frame r_t and the corresponding true 3D bounding box r_a , then the overlap fraction S can be defined as

$$S = \frac{|r_i \cap r_a|}{|r_i \cup r_a|} \quad (9)$$

The intersection and union between two regions are denoted by $\cap$ and $\cup$, respectively, while $|\cdot|$ represents the number of points in a region. A frame is considered successful if the overlap S between the predicted and ground truth bounding boxes is greater than a set threshold t_0 , and the success rate is the percentage of successful frames over all frames. Precision is the percentage of frames where the distance between the center of the predicted bounding box and the center of the ground truth bounding box is less than a given threshold. The Area Under the Curve (AUC) is used to evaluate the tracking performance with a maximum threshold ranging from 0 to 1. Accuracy is the AUC with a maximum error of 0 to 2 m.

3.3 Comparison with state-of-the-art approaches

Results on KITTI. The MUT model was evaluated on the KITTI dataset^[12] and compared with 10 state-of-the-art methods. The results are presented in Table 1. It is evident from the table that the MUT outperforms all other methods across all categories. When compared with LTTR^[33] and PTTR^[34] that utilize the Transformer module, the MUT model exhibits a 2.5% to 30.8% improvement in the Car and Pedestrian categories, which have the largest number of training samples. The substantial improvement in the Van and Cyclist categories, despite their limited number of samples, demonstrates that the MUT model is capable of achieving satisfactory results with fewer training samples. Furthermore, compared to M2Track^[6], the MUT model

exhibits significant performance gains in all categories.

As shown in Fig. 5, when the number of object points is sufficient, MUT is closer to the ground truth than M2Track^[7]. Even when the object points become sparse to the point where even humans cannot directly identify which points belong to the object point cloud, MUT can still obtain convincing tracking results. At the same time, M2Track^[7] suffers a significant setback in position and offset angle prediction, with a large difference between the predicted bounding box and the ground truth. This is not surprising, since as the point cloud becomes sparser, the uncertainty in the center point prediction increases, and so does the randomness of the model predictions. The uncertainty in the localization of the center coordinates affects the parameters of the model, and once the number of object points is greatly reduced, the network will experience performance loss. However, if we introduce uncertainty modeling, it can learn the true distribution of the data and obtain good generalization performance, enabling the model to be applicable even in extreme cases.

Results on NuScenes. We compared our proposed model with four open-source methods, namely, SC3D^[14], P2B^[7], BAT^[28], and M2Track^[6]. The results are shown in Table 2. Our model also outperforms these methods in all categories. Compared to the KITTI dataset^[12], the NuScenes dataset^[13] is more challenging. The dataset contains scenes with severe interference and drastic appearance changes, making object localization extremely difficult. P2B^[7] and BAT^[28] both follow the appearance-matching paradigm, so scenes with severe interference and drastic appearance changes will confuse the model's object recognition, leading to a decrease in network performance during matching. The M2Track^[6], on the

Table 1 Results of a comparison between our MUT and the state-of-the-art methods on the KITTI dataset (red indicates the best performance, and blue indicates second-best performance). (%)

Metric	Method	Car (6424 frames)	Pedestrian (6088 frames)	Van (1248 frames)	Cyclist (308 frames)	Mean (14 068 frames)
Success rate	SC3D ^[14]	41.3	18.2	40.4	41.5	31.2
	SC3D-RPN ^[31]	36.3	17.9	-	43.2	-
	P2B ^[7]	56.2	28.7	40.8	32.1	39.5
	3D-SiamRPN ^[15]	58.2	35.2	45.7	36.2	43.8
	SA-P2B ^[32]	58.0	34.6	51.2	32.0	44.0
	MLVSNet ^[17]	56.0	34.1	52.0	34.3	44.1
	BAT ^[28]	60.5	42.1	52.4	33.7	47.2
	LTTR ^[33]	65.0	33.2	35.8	66.2	50.0
	PTTR ^[34]	65.2	50.9	52.5	65.1	58.4
	M2Track ^[6]	65.5	61.5	53.8	73.2	62.9
Precision	Ours	67.7	61.0	58.3	75.6	63.0
	SC3D ^[14]	57.9	37.8	47.0	70.4	48.5
	SC3D-RPN ^[31]	51.0	47.8	-	81.2	-
	P2B ^[7]	72.8	49.6	48.4	44.7	53.9
	3D-SiamRPN ^[15]	76.2	56.2	52.9	49.0	58.6
	SA-P2B ^[32]	75.1	63.3	63.1	43.6	61.3
	MLVSNet ^[17]	74.0	61.1	61.4	44.5	60.3
	BAT ^[28]	77.7	70.1	67.0	45.4	65.1
	LTTR ^[33]	77.1	56.8	45.6	89.9	67.4
	PTTR ^[34]	77.4	81.6	61.8	90.5	77.8
	M2Track ^[6]	80.8	88.2	70.7	93.5	83.4
	Ours	81.3	87.7	74.0	95.0	83.8

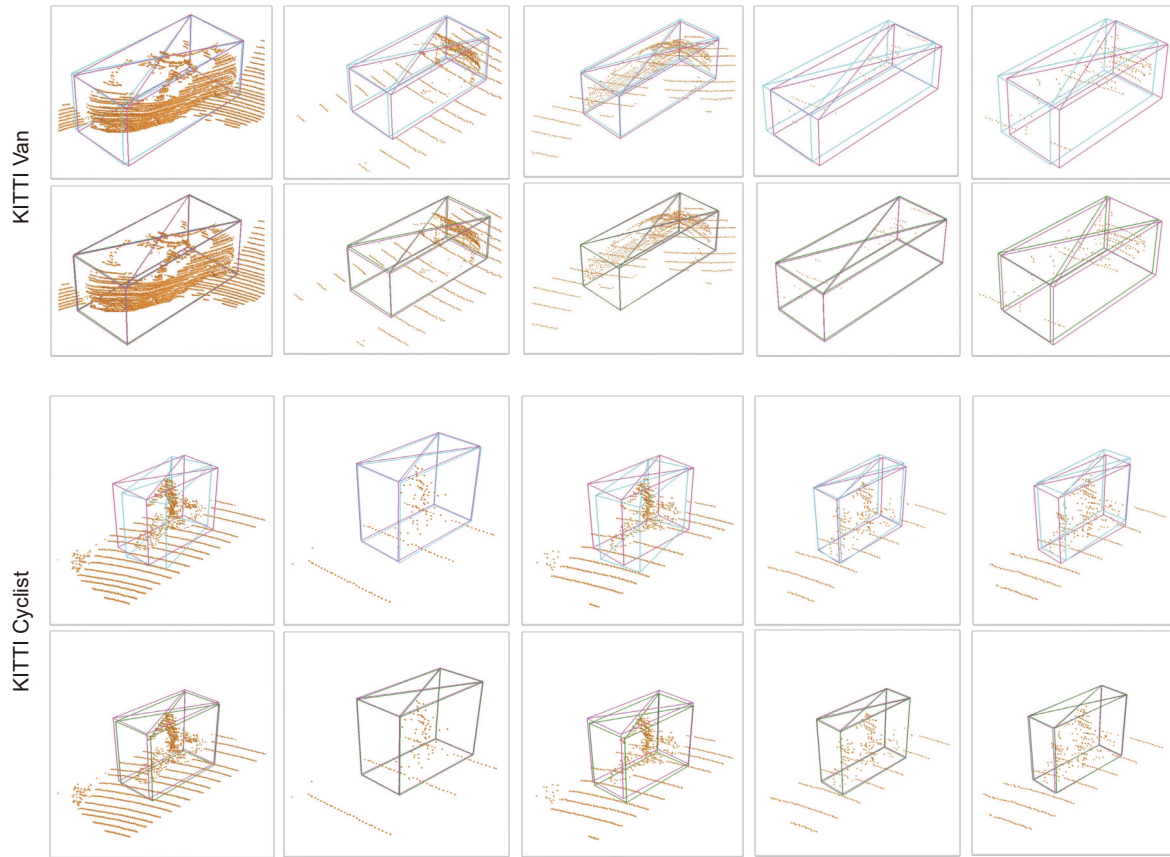


Fig. 5 Visualization results. Red for ground truth, blue for M2Track, and green for Ours.

Table 2 Results of a comparison between our MUT and the state-of-the-art methods on the NuScenes dataset^[13] (red indicates the best performance, and blue indicates second-best performance). (%)

Metric	Method	Car (64 159 frames)	Truck (13 587 frames)	Trailer (3352 frames)	Bus (2953 frames)	Mean (84 051 frames)
Success rate	SC3D ^[14]	22.3	30.7	35.3	29.4	29.4
	P2B ^[7]	38.8	43.0	49.0	33.0	41.0
	BAT ^[28]	40.7	45.3	52.6	35.4	44.4
	M2Track ^[6]	55.8	57.3	57.6	51.3	49.2
	Ours	57.0	59.6	62.9	58.9	55.3
Precision	SC3D ^[14]	21.9	27.7	28.1	24.1	25.5
	P2B ^[7]	43.2	41.6	40.1	27.4	38.1
	BAT ^[28]	43.3	42.6	44.9	28.0	40.0
	M2Track ^[6]	65.1	59.5	58.3	51.4	62.7
	Ours	65.3	62.2	61.2	57.9	64.1

other hand, tracks objects using motion information. However, it cannot achieve satisfactory performance in scenarios with small training samples and large objects. Our model uses uncertainty modeling to consciously learn the true central coordinate distribution by adding heteroscedastic estimates for severely disturbed objects. Combined with efficient knowledge transfer and holistic and detailed descriptions of point cloud scenes, the model can perform well even in extreme cases. For large objects such as trucks, trailers, and buses, our model shows improvements of about 3%–7% compared to M2Track^[7]. Even in the complex Car category, the model has an improvement of 0.2% and 1.2% in precision and success rate, respectively. This indicates that our proposed method is highly effective.

3.4 Analysis experiments

This section analyzes how each component of the MUT affects it and explores the factors that drive tracker performance. Unless stated otherwise, the experiments presented in this section are carried out on the Van category of the KITTI dataset^[12].

Ablations. We conducted detailed ablation experiments on the KITTI dataset^[12] to investigate the impact of each component of MUT on performance. Specifically, we ablated the HDD, MVFE, and uncertainty modeling modules, and the results are shown in Table 3. The HDD component improves the performance of the baseline module by about 1.5%, indicating that using both the holistic and detailed descriptors of point clouds can provide the network with a more comprehensive understanding of the

Table 3 Results of MUT when different components are ablated, and finally represents the full model. Red represents the best performance. (%)

Component				Success rate	Precision
Baseline	HDD	MVFE	Uncertainty modeling		
√	-	-	-	53.8	70.7
√	√	-	-	55.0	71.7
√	√	√	-	56.1	72.3
√	√	√	√	58.3	74.0

tracking scene.

The improvement from the MVFE component was slightly smaller than that from the HDD module, with an improvement of less than 1%. This is because the HDD module provides the model with a complete understanding of the point cloud scene, and the information provided by the depth map feature does not further improve network performance. Nevertheless, the depth map feature can still assist the network in the case of sparse point clouds, resulting in a decent improvement. The improvement in uncertainty modeling is significant. This is mainly because uncertainty modeling adds heteroscedasticity estimation to the center coordinate positioning, assuming that the data distribution is inconsistent so that the model can learn the real distribution. This approach also allows the tracker to focus more on optimizing bounding boxes with large prediction biases. Overall, we find that each module can effectively improve network performance.

Factors that limit performance. In this section, we will carefully examine the limiting factors that affect the 3D SOT model. As shown in Table 4, when we replace the predicted center coordinates of the BBox with the ground-truth center coordinates, the network achieves a 48.7% improvement in success rate and a 36.9% improvement in precision. However, when we replace the predicted angle of the BBox with the ground-truth angle, the performance only improves by less than 10%. This indicates that the inability to accurately predict the center coordinates of the BBox is a key factor limiting the performance improvement of the network. Although the model can predict the angle of the object well, it cannot accurately predict the center coordinates due to sparse object points, sampling, and regression strategies. The original baseline model directly regressed the center coordinates and angle, but the performance was poor.

When we decoupled them and used the center detector and angle detector to predict them separately, without any other changes, the performance improved by about 2%. This is mainly because the center coordinates and angle do not belong to the same solution space, and regressing them together may lead to catastrophic results. At the same time, this also means that the predicted center coordinates of the network have a high degree of uncertainty. By adding uncertainty estimation to the baseline model, we can see that the performance improves by about 5%.

Table 4 Factors affecting tracking performance. (%)

Module	Success rate	Precision
Baseline	48.7	63.1
Add ground truth center	91.6	100.0
Add ground truth angle	57.1	63.1
Add decouple	51.3	64.7
Add uncertainty modeling	57.1	69.3

This fully demonstrates that uncertainty modeling can enable the model to learn the true central coordinate distribution and optimize poor coordinate estimates.

Choice of the activation function. Variance is a statistical measure that describes the spread of a dataset around its mean. It is computed by taking the average of the squared deviations of each data point from the mean. In machine learning, it is desirable to train a network that can learn the variance of a dataset. However, Conventional Neural Networks (CNNs) or Multilayer Perceptrons (MLPs) are not capable of ensuring that the learned variance, denoted as σ , is strictly positive. To overcome this limitation, activation functions can be leveraged to enforce the positivity constraint on the learned variance σ .

In our study, we conducted experiments with four different activation functions to address the challenge of ensuring that the learned variance is greater than zero. The experimental results are presented in Table 5. Our observations indicate that using the absolute function is the simplest way to achieve this objective. However, it is also the least effective method as it is not differentiable, which can severely impede network training. The commonly used Softmax and ReLU functions also show poor performance as they hinder network convergence. In contrast, the sigmoid function is the most effective as it constrains the variance (σ) to the range $[0, 1]$ and is differentiable, enabling the network to modify the value of σ more rapidly. Based on our findings, we recommend using the sigmoid function as an activation function in neural networks to ensure that the learned variance is positive. This approach has been demonstrated to be effective in facilitating the training of the network.

Influence of HDD. In this section, we investigated the impact of holistic and detailed descriptors on the tracking network. The experimental results are presented in Table 6. The results show that using only detailed descriptors of the point cloud leads to a performance drop of about 4% compared to the default setting. This indicates that using only detailed descriptors of the point cloud is not sufficient for the network to fully understand the point cloud scene. This is because the information on the object point will inevitably be lost during the downsampling process. If the originally collected point cloud is incomplete, this approach will undoubtedly lead to a sharp drop in network performance. Compared to using only holistic descriptors, using only detailed descriptors produces a performance difference of about 3%. This clearly shows that using only holistic descriptors may be better

Table 5 Effect of different activation functions on variance σ . Red represents the best performance. (%)

Activation function	Success rate	Precision
Abs	48.8	63.1
Softmax	54.1	70.2
ReLU	50.6	65.3
Sigmoid	55.0	71.7

Table 6 Influence of holistic and detailed descriptors on the tracking network. Red represents the best performance. (%)

Module	Success rate	Precision
Holistic descriptors	53.8	70.7
Detailed descriptors	52.8	67.5
HDD	55.0	71.7

because holistic descriptors contain information about the entire scene. However, holistic descriptors lack detailed descriptions, which still leads to a loss in network performance. When the model simultaneously uses holistic and detailed descriptors for tracking, both precision and success rate are significantly improved. This highlights the fact that both holistic and detailed descriptive information of the point cloud scene is essential for tracking and cannot be replaced by one another.

Comparison of two sampling strategies. In this section, we re-examine the Farthest Point Sampling (FPS) strategy commonly used by the backbone PointNet++^[11] in the 3D SOT field. Undoubtedly, the FPS strategy is not very suitable for 3D SOT models due to its enormous computational complexity and point sampling that may cause missing object information. As shown in Table 7, the computational cost of the FPS strategy is about 70% higher than that of our Former Attraction Probability Sampling (FAPS) strategy, but its precision decreases by about 8%. This is because the FPS strategy retains a large number of background points, which is fatal for 3D SOT models. After all, it causes significant displacement in the prediction of the center coordinates of the model. On the other hand, our FAPS strategy tends to retain object points, which leads to less loss of object information during downsampling. Moreover, without tedious loop processes, we can have less computational cost and faster inference speed, making the network more practical for real-world use.

Comparison of training costs. In this chapter, we further investigate the training cost (we fixed all the model parameters of CLIP^[35]). We chose the NuScenes dataset^[13] as the experimental data. Table 8 shows that, compared with P2B^[7], our model increased the FLOPs and parameters by 1.7×10^9 and 5.5×10^6 , respectively, but also improved the success rate and precision by 14.3% and 26.0%. In addition, we find that P2B^[7], having the least number of parameters, has much higher FLOPs than M2Track^[6]. This is mainly because P2B^[7] used the original PointNet++^[11] to extract features. PointNet++^[11], which uses farthest point sampling, has no parameters, but has a high computational cost. M2Track^[6] replaced PointNet++^[11] with a mini MLP structure, which increased some parameters (mainly due to the segmentation network), but did not significantly increase the computational cost. Our model, because it uses both mini MLP and modified PointNet++^[11], has a certain increase in FLOPs and parameters. But because we used FAPS to replace the farthest point sampling strategy, our FLOPs increase is less than the parameter increase. Also, MUT has the best results among the three on the NuScenes dataset^[13]. This is acceptable in terms of performance and

efficiency.

4 Conclusion

In this work, we propose MUT, a novel multi-view unsupervised center uncertainty 3D single object tracker. The proposed method consists of multiple components: uncertainty modeling, multi-view visual feature extractor, and holistic and detailed descriptor. Each of these components is designed to compensate for specific limitations in object tracking. The uncertainty modeling helps the model learn the center coordinate distribution, leading to more robust predictions for 3D BBox. The multi-view visual feature extractor helps the model understand the object shape and achieve efficient knowledge transfer. The integration of both holistic and detailed descriptors for point clouds also compensates for their respective limitations, leading to improved performance. While each component can individually improve tracking performance, their combination in a unified framework yields better performance. Extensive experimental results demonstrate that MUT outperforms previous state-of-the-art techniques and achieves satisfactory tracking performance even in extreme cases.

Article History

Received: 13 March 2023; Revised: 26 April 2023; Accepted: 14 June 2023

References

- [1] E. Machida, M. Cao, T. Murao, and H. Hashimoto, Human motion tracking of mobile robot with Kinect 3D sensor, in *Proc. 2012 SICE Annu. Conf. (SICE)*, Akita, Japan, 2012, pp. 2207–2211.
- [2] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, pp. 3101–3109, 2021.
- [3] T. Yin, X. Zhou, and P. Krähenbühl, Center-based 3D object detection and tracking, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 11779–11788.
- [4] A. I. Comport, E. Marchand, and F. Chaumette, Robust model-based tracking for robot vision, in *Proc. 2004 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2005, pp. 692–697.
- [5] L. Landrieu and M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4558–4567.
- [6] C. Zheng, X. Yan, H. Zhang, B. Wang, S. Cheng, S. Cui, and Z. Li, Beyond 3D Siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 8101–8110.
- [7] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, P2B: Point-to-box network for 3D object tracking in point clouds, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 6328–6337.
- [8] C. R. Qi, O. Litany, K. He, and L. Guibas, Deep Hough voting for 3D object detection in point clouds, in *Proc. 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Seoul, Republic of Korea, 2020, pp. 9276–9285.
- [9] X. Ma, Y. Zhang, D. Xu, D. Zhou, S. Yi, H. Li, and W. Ouyang, Delving into localization errors for monocular 3D object detection, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 4719–4728.
- [10] Y. Zhang, Q. Zhang, Z. Zhu, J. Hou, and Y. Yuan, GLENet: boosting 3D object detectors with generative label uncertainty

Table 7 Impact of different sampling strategies on performance. The sampling strategies include FPS and FAPS. Red represents the best performance.

Sampling strategy	Success rate (%)	Precision (%)	FLOP	Number of parameters
FPS	51.3	64.7	9.5×10^9	7.1×10^6
FAPS	56.4	73.0	5.6×10^9	6.7×10^6

Table 8 Comparison of training parameters (experimental results based on NuScenes dataset^[13], and red indicates best performance).

Method	FLOP	Number of parameters	Success rate (%)	Precision (%)
P2B ^[7]	4.0×10^9	1.2×10^6	41.0	38.1
M2Track ^[6]	3.1×10^9	2.4×10^6	49.2	62.7
Ours	5.7×10^9	6.7×10^6	55.3	64.1

- estimation, arXiv preprint arXiv: 2207.02466, 2022.
- [11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, arXiv preprint arXiv: 1706.02413, 2017.
 - [12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, Vision meets robotics: The KITTI dataset, *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
 - [13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, NuScenes: A multimodal dataset for autonomous driving, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 11618–11628.
 - [14] S. Giancola, J. Zarzar, and B. Ghanem, Leveraging shape completion for 3D Siamese tracking, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2020, pp. 1359–1368.
 - [15] Z. Fang, S. Zhou, Y. Cui, and S. Scherer, 3D-SiamRPN: An end-to-end learning method for real-time 3D single object tracking using raw point cloud, *IEEE Sens. J.*, vol. 21, no. 4, pp. 4995–5011, 2021.
 - [16] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, High performance visual tracking with Siamese region proposal network, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 8971–8980.
 - [17] Z. Wang, Q. Xie, Y. K. Lai, J. Wu, K. Long, and J. Wang, MLVSN: Multi-level voting Siamese network for 3D visual tracking, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Montreal, Canada, 2022, pp. 3081–3090.
 - [18] A. Kendall and Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, California, USA, 2017, pp. 5580–5590.
 - [19] J. Choi, D. Chun, H. Kim, and H. J. Lee, Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving, in *Proc. 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Seoul, Republic of Korea, 2019, pp. 502–511.
 - [20] Y. Zhang, J. Lu, and J. Zhou, Objects are different: Flexible monocular 3D object detection, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 3288–3297.
 - [21] H. Chen, Y. Huang, W. Tian, Z. Gao, and L. Xiong, MonoRUN: monocular 3D object detection by reconstruction and uncertainty propagation, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 10374–10383.
 - [22] J. Yang, J. Duan, S. Tran, Y. Xu, S. Chanda, L. Chen, B. Zeng, T. Chilimbi, and J. Huang, Vision-language pre-training with triple contrastive learning, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 15650–15659.
 - [23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv: 2010.11929, 2020.
 - [24] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.
 - [25] R. Q. Charles, S. Hao, K. Mo, and L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 77–85.
 - [26] D. A. Nix and A. S. Weigend, Estimating the mean and variance of the target probability distribution, in *Proc. 1994 IEEE Int. Conf. Neural Networks (ICNN'94)*, Orlando, FL, USA, 2002, pp. 55–60.
 - [27] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, arXiv preprint arXiv: 1506.01497, 2015.
 - [28] C. Zheng, X. Yan, J. Gao, W. Zhao, W. Zhang, Z. Li, and S. Cui, Box-aware feature enhancement for single object tracking on point clouds, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision (ICCV)*, Montreal, Canada, 2022, pp. 13179–13188.
 - [29] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980, 2014.
 - [30] Y. Wu, J. Lim, and M. H. Yang, Online object tracking: A benchmark, in *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, USA, 2013, pp. 2411–2418.
 - [31] J. Zarzar, S. Giancola, and B. Ghanem, Efficient bird eye view proposals for 3D Siamese tracking, arXiv preprint arXiv: 1903.10168, 2019.
 - [32] X. Zhou, L. Wang, Z. Yuan, K. Xu, and Y. Ma, Structure aware 3D single object tracking of point cloud, *J. Electron. Imag.*, vol. 30, no. 4, p. 043010, 2021.
 - [33] Y. Cui, Z. Fang, J. Shan, Z. Gu, and S. Zhou, 3D object tracking with transformer, arXiv preprint arXiv: 2110.14921, 2021.
 - [34] C. Zhou, Z. Luo, Y. Luo, T. Liu, L. Pan, Z. Cai, H. Zhao, and S. Lu, PTTR: Relational 3D point cloud object tracking with transformer, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 8521–8530.
 - [35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, arXiv preprint arXiv: 2103.00020, 2021.