Enhancing 3D Object Tracking via Dual-Context Propagation and Temporal Context Fusion

Peijing Jiang[1] Yuanping Zhang[1*] Jinlong Pang[1] Zhongjun Lin[1]
[1]Southwest University

Abstract

Point cloud-based 3D single object tracking (3D SOT) plays a pivotal role in applications such as autonomous driving and robotic vision. Despite recent progress, most existing approaches rely solely on current-frame features for target localization. This approach overlooks temporal information that is crucial for robust tracking under occlusion, appearance variations, and sparse point clouds. In addition, the effectiveness of 3D SOT largely depends on the quality of feature fusion between the target template and the search region. Traditional fusion strategies often suffer from limited interaction capacity and weak discriminative representation. To address these challenges, we propose DT-Tracker, which performs multi-layer bidirectional feature interaction and temporal cue propagation to improve tracking robustness and feature discrimination capability. Specifically, we introduce a Dual-Context Propagation Network that applies bidirectional cross-attention across multiple layers between the template and search region, enabling deep semantic alignment and progressive feature refinement. Furthermore, we design a Temporal Context Fusion module that adaptively incorporates temporal cues from historical fusion features into the current frame, effectively improving resilience to occlusion and appearance drift. Extensive experiments on the KITTI and nuScenes datasets demonstrate that DT-Tracker achieves competitive results compared to existing representative methods.

1. Introduction

3D Single Object Tracking (3D SOT) has garnered increasing attention due to its wide applicability in fields such as autonomous driving [15, 18], visual surveillance [19], augmented reality [26] and robotic vision [2]. With the proliferation of 3D sensors like LiDAR, point cloud-based 3D SOT methods [8, 23, 29, 33, 34] have become a research hotspot recently. The task aims to continuously estimate the target's position in each subsequent frame, given its initial state.

*Corresponding author.

Among existing approaches, the Siamese network-based appearance matching paradigm [8, 10, 21, 23, 33] is widely adopted for its structural simplicity and stable performance. SC3D [8] is a representative method that pioneered the use of 3D Siamese architectures for proposal-based matching. P2B [23] further improved tracking efficiency and accuracy by employing end-to-end training along with a Hough voting scheme [22]. However, most of these methods rely solely on the current frame for target localization, ignoring valuable temporal information from previous frames. This limitation renders them less robust in challenging scenarios such as occlusion, appearance variations, and sparse observations. On the other hand, motion-centric methods focus on inter-frame dynamics. For example, M2-Track [34] segments target point clouds from two consecutive frames using prior predictions, applies offset regression for coarse prediction, and subsequently refines the results. While this strategy is effective in dynamic environments, it may be vulnerable to displacement variations across datasets and shows reduced efficiency when tracking large-scale objects [29]. More recently, efforts have emerged to incorporate temporal modeling into 3D tracking. STDA [30] introduces a temporal completion module, but its reliance on external detectors hinders end-to-end optimization and results in discontinuous template matching. STTracker [5] aggregates historical features in Bird's Eye View (BEV) space to improve robustness to occlusion and deformation. Nonetheless, it depends on voxelized representations and treats all historical frames equally, which neglects finer point-level structures and temporal saliency.

Meanwhile, feature fusion remains a critical component of 3D SOT systems. Approaches such as P2B [23] and PTT [24] utilize cosine similarity for target-guided fusion, while methods like BAT [33], DMT [28], and GLT-T [21] incorporate geometric information from bounding boxes. Despite their success, most existing methods adopt unidirectional fusion—typically using template features to guide the search branch—while ignoring reciprocal guidance. Moreover, their interactions often occur only at shallow levels, limiting semantic complementarity and contextual depth. Transformer-based trackers, such as LTTR [3],

PTTR [35], Inception-Track [17], and OSP2B [20], employ cross-attention to align features between the template and search regions. However, their fusion is often restricted to early network layers, where features carry limited semantic information. This shallow-level fusion tends to repeatedly integrate low-level patterns, resulting in redundant information and underutilization of richer semantic context from deeper layers. STNet [11] introduces iterative cross-attention for deeper interaction, but it still lacks adaptive refinement of template features. Similarly, MCSTN [6] performs multi-level correlation-based enhancement. However, it is unable to effectively enable bidirectional interaction, as the process of voxelization leads to a loss of resolution

To address these limitations, we propose DT-Tracker, a novel 3D tracker that integrates a Dual-Context Propagation Network (DCP-net) and a Temporal Context Fusion (TCF) module. Specifically, we begin by extracting geometric-semantic features from the template and search point clouds using a shared-weight backbone. Then, DCP-net performs multi-layer bidirectional cross-attention, allowing mutual guidance between template and search features at each layer. After that, the TCF module captures temporal dependencies by propagating informative cues from historical fusion features to the current frame, which enables the tracker to better handle occlusion, appearance changes, and spatial continuity across frames. Finally, the refined and discriminative fused features are passed to a target localization head for final prediction. Extensive experiments on the KITTI [7] and nuScenes [1] benchmarks confirm the effectiveness and generalization of our approach.

In summary, our main contributions are: (1) We propose an end-to-end 3D single object tracker, DT-Tracker, which effectively leverages temporal information and attention mechanisms to mitigate performance degradation caused by significant target appearance changes and insufficient feature fusion. (2) We design a Dual-Context Propagation Network that simultaneously performs cross-guidance between template and search region at each layer and densely aggregates features across layers, generating discriminative target-specific fusion features. This module significantly improves deep semantic consistency and structural alignment in feature fusion. (3) We design a Temporal Context Fusion module that integrates target feature cues from previous frames into the current search region features via cross-attention, enhancing the model's temporal awareness and adaptability to target motion changes.

2. Related Work

2.1. Appearance-Based 3D Object Tracking

Appearance-based tracking methods rely on comparing semantic features between the template and search regions, typically using Siamese architectures. Early works like SC3D [8] propose using shape completion to regularize candidate-template matching. P2B [23] introduces a point-wise matching scheme and employs Hough voting [22] for target center estimation. PTT [24] enhances P2B by injecting self-attention into the VoteNet-based detection head for improved local context modeling. BAT [33] proposes a box-aware feature that encodes point-to-box distances to capture instance-specific geometric cues. V2B [10], in contrast, does not follow the point-wise voting strategy. Instead, it voxelizes point clouds and maps them to a BEV (Bird's Eye View) representation for target localization via a voxel-to-BEV regression head. Similarly, LTTR [3] separately encodes template and search point clouds into BEV spaces and fuses them using Transformer layers to enhance long-range spatial context modeling. Recent works further improve template-search interaction. STNet [11] constructs an encoder-decoder network using self-attention and cross-attention mechanisms, and employs an attention mechanism to capture the robust cross-correlation between the template and search regions. GLT-T [21] redesigns the vote proposal strategy to overcome limitations of single-seed guidance in VoteNet [22]. Li et al. [16] introduce a pre-segmentation module that incorporates bounding box size cues and leverages both siamese cross-attention and self-attention mechanisms to suppress background interference during feature encoding. Departing from the two-stream structure, OST [32] concatenates template and search point clouds and feeds them jointly into a shared Transformer, enabling the model to extract interaction-aware features in a unified encoding process without requiring explicit correlation modules.

Our DT-Tracker follows the appearance matching paradigm, but extends it by incorporating temporal context and dual-directional attention, enhancing feature discriminability under challenging scenarios such as occlusion, sparsity, and appearance drift.

2.2. Motion-Based 3D Object Tracking

Motion-based tracking methods leverage temporal cues across frames to improve robustness in scenarios involving occlusion, sparsity, or appearance changes. In contrast to appearance-only paradigms, these approaches explicitly model the dynamics of the target object. M2-Track [34] addresses motion by segmenting targets from two consecutive frames and applying offset regression with a refinement step. SETD [25] presents a unified framework that couples target discrimination with state estimation. It extends the Lucas-Kanade algorithm to 3D tracking by learning incremental warp parameters through a deep network, enabling continuous refinement of target states. DMT [28] introduces a lightweight, detector-free tracker that relies purely on motion prediction. It estimates a coarse target center

using previous states and refines the result through an explicit voting module, achieving efficient and accurate tracking without 3D detection heads. RDT [12] approaches motion modeling from a registration perspective. By predicting a rigid transformation between the template and search areas, it spatially aligns point clouds before feature aggregation. This registration-driven design enhances tracking accuracy by ensuring geometric consistency across frames, particularly in ambiguous or sparse conditions.

While the adoption of motion cues significantly improves robustness, this paradigm still faces challenges. It remains difficult to accurately predict target motion within sparse and noisy point clouds [32], and performance may degrade under large displacements or scale variations [29].

2.3. Feature Interaction Strategies

Feature fusion plays a pivotal role in determining 3D SOT performance. Recent works have sought to enhance the interaction between the template and search regions. For instance, P2B [23] and PTT [24] employ cosine similarity for target-guided fusion, while BAT [33] and GLT-T [21] incorporate geometric priors derived from bounding box cues. However, their reliance on handcrafted similarity metrics or static geometric encodings limits their capacity to model complex interactions, making them less adaptable to appearance variations, occlusions, or spatial ambiguity. Moreover, compared to Transformer-based methods, these conventional strategies lack sufficient capacity for modeling long-range dependencies or multi-level semantics. To overcome these limitations, Transformer-based methods introduce more expressive attention mechanisms. Trackers such as LTTR [3], PTTR [35], Inception-Track [17], and OSP2B [20] leverage cross-attention to enhance feature alignment, though often restricted to early layers—potentially leading to information redundancy and underutilized deep context. STNet [11] enhances this by employing iterative cross-attention across layers, but still lacks adaptive mechanisms for template feature refinement. MCSTN [6] improves multi-level correlation using voxel-based attention, yet fails to effectively enable bidirectional interaction due to resolution loss from voxelization.

To address these limitations, we propose the Dual-Context Propagation Network, which performs feature fusion via multi-layer bidirectional cross-attention for mutual guidance between template and search features. This design enables deep symmetric interaction and improves the discriminability of fused features. Ablation studies confirm the advantage of bidirectional attention in modeling robust point cloud relationships for tracking.

2.4. Temporal Modeling in Point Clouds

While appearance-based tracking methods mainly rely on current-frame features, robust performance in complex scenarios increasingly demands effective temporal modeling. Recently, work has started to attempt to introduce temporal modeling into 3D SOT. STDA [30] introduces a temporal motion completion module, but its reliance on external detectors hinders end-to-end training, and its direct jump from the initial template to the current frame compromises temporal continuity. STTracker [5] fuses multiple historical BEV features to enhance robustness under occlusion and appearance change. However, it discards point-level structural detail and treats all history frames with equal weight, limiting adaptiveness.

To address these issues, we introduce the Temporal Context Fusion module. The TCF module leverages historical fusion features to extract and propagate temporal cues via cross attention, adaptively emphasizing informative past frames. This enables better handling of occlusion, appearance drift, and structural variation. By incorporating temporal context into current-frame representations, our model enhances long-sequence coherence and improves robustness under dynamic conditions.

3. Proposed Method

Problem Definition. Given a 3D point cloud template and a sequence of search region point clouds, the goal of 3D single object tracking is to localize the target by estimating its 3D bounding box $\{x, y, z, l, w, h, \theta\}$ in each frame. The tracker receives the initial state and point cloud of the target in the first frame and must continuously estimate its state in subsequent frames.

3.1. Overview of DT-Tracker

Fig. 1 (a) illustrates the overall architecture of DT-Tracker. In point cloud-based 3D SOT, the target state is typically represented by a 3D bounding box parameter set $\{x, y, z, l, w, h, \theta\} \in \mathbb{R}^7$, where (x, y, z) denotes the center, (l, w, h) denotes the dimensions, and θ denotes the orientation angle. Since the target's 3D size remains constant or changes minimally across frames, only $(x, y, z, \theta) \in \mathbb{R}^4$ need to be estimated. Given the target's initial state and corresponding template point cloud (N_t points), we predict its state within the search region point cloud (N_s points) of each frame.

As shown in Fig. 1 (a), DT-Tracker first employs a shared-parameter Siamese Point Transformer backbone, as in [11], to extract rich geometric and semantic features from the template and search point clouds, yielding template features $\mathbf{F}_t \in \mathbb{R}^{N_t \times C}$ and search features $\mathbf{F}_s \in \mathbb{R}^{N_s \times C}$. Specifically, we fuse both geometric features that capture local shape structures and semantic features that encode high-level appearance information from the template and search regions through bidirectional attention. These fused representations are then input to the Dual-Context Propagation Network (DCP-net). Within DCP-net, features are

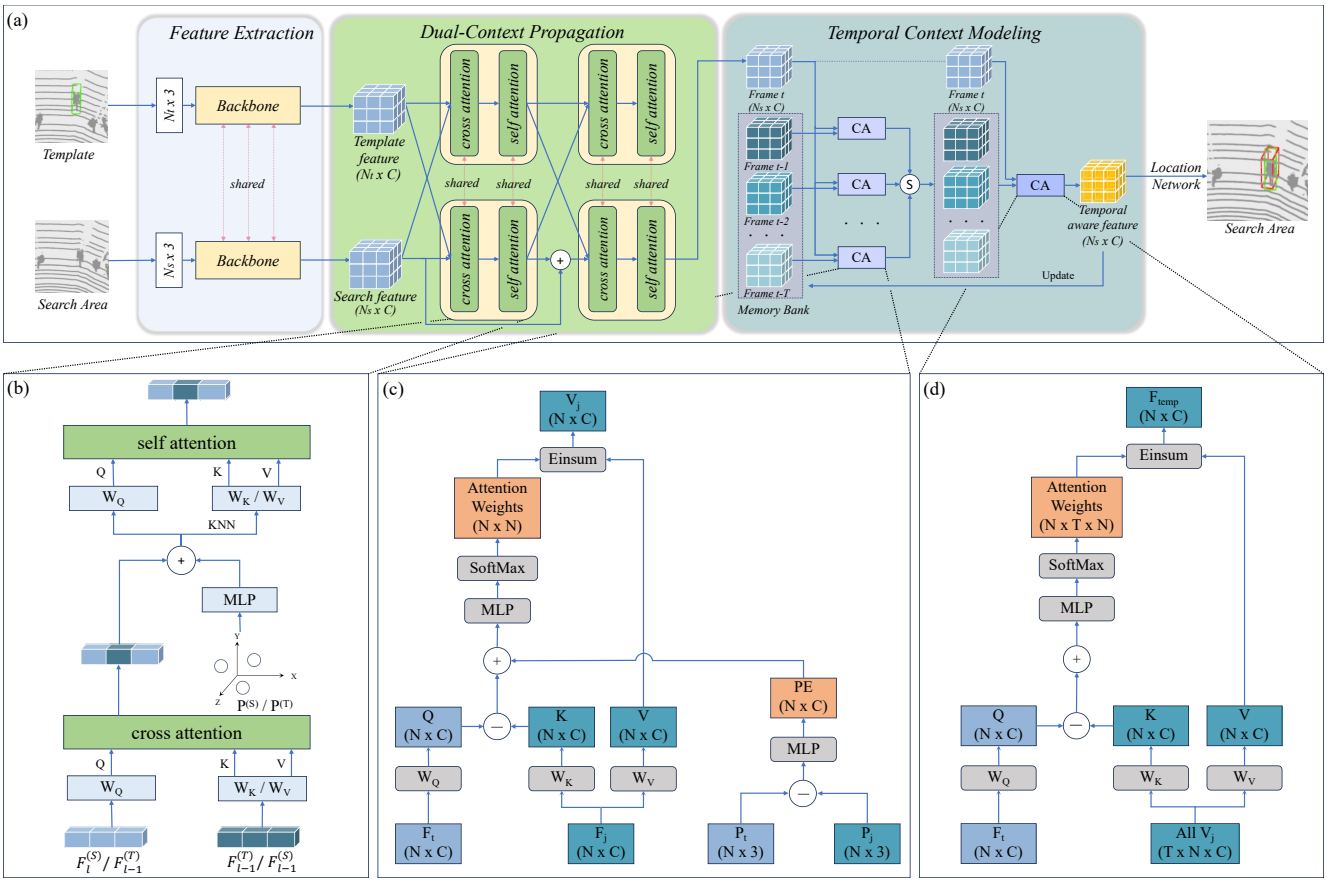

Figure 1. **The overall architecture of the DT-Tracker.** First, there is a shared-weight backbone that extracts geometric-semantic features from the template point cloud and search region point cloud. Then, the Dual-Context Propagation Network comes into play, which performs multi-layer bidirectional cross-attention (enabling mutual guidance between template and search features) and local self-attention (LSA) for progressive feature refinement and structural alignment. After that, the Temporal Context Fusion module is used, which aggregates historical fused features from T previous frames via cross-frame attention to enhance temporal continuity and handle occlusion/appearance changes. Finally, the refined features are fed to a localization head for 3D bounding box prediction. "+" indicates element-wise addition, "S" denotes stacking, and "–" represents channel-wise subtraction.

updated via layer-wise bidirectional cross-attention, followed by Local Self-Attention (LSA) to capture local context. To further enhance temporal consistency and dynamic compensation, we introduce the Temporal Context Fusion (TCF) module after DCP-net. The TCF module takes the current frame's fused features and corresponding coordinates, aligns and fuses them with features and coordinates from the previous T frames, outputting temporally enhanced features. Finally, these features are fed into a localization head to predict the target's 3D bounding box.

3.2. Feature Extraction Backbone

To extract discriminative features from sparse point clouds, we adopt the Siamese Point Transformer backbone from STNet [11]. This encoder-decoder architecture incorporates attention mechanisms to jointly capture fine-grained local geometries and holistic shape context.

As illustrated in Fig. 1, the backbone processes the template and search region point clouds via two shared-weight branches. The network hierarchically downsamples each input to extract multi-scale features and then applies upsampling layers to recover high-resolution representations. The final outputs are the template features $\mathbf{F}_t \in \mathbb{R}^{N_t \times C}$ and search features $\mathbf{F}_s \in \mathbb{R}^{N_s \times C}$, which encode both local geometry and global context, providing a rich representation for subsequent matching and localization tasks.

3.3. Dual-Context Propagation Network

Effective feature interaction between template and search regions is crucial for accurate 3D tracking, as it enables the network to leverage semantic correspondences for robust localization. However, existing methods often process the two branches independently, missing important contextual alignment and mutual guidance. To address this, we pro-

pose the Dual-Context Propagation Network. Given template features $\mathbf{F}_t$ and search features $\mathbf{F}_s$ with their coordinates $\mathbf{P}^{(T)}$ and $\mathbf{P}^{(S)}$, DCP-net enhances cross-region interaction through three modules: Dense Feature Propagation to retain low-level geometric details, Bidirectional Cross-Attention for mutual feature guidance, Local Refinement to preserve spatial consistency.

Deep interaction in point-based networks often leads to the loss of fine-grained geometric details due to the progressive weakening of spatial structure. To alleviate this, we apply dense connection [6] which inspired by [9] in the search branch by summing previous layer outputs. Specifically, the search input at layer l is computed as:

$$\mathbf{F}_l^{(S)} = \sum_{i=0}^{l-1} \mathbf{F}_i^{(S)} \tag{1}$$

where $\mathbf{F}_0^{(S)} = \mathbf{F}_s$. This design allows geometric cues from shallow layers to directly influence deeper representations, mitigating information loss and promoting structural consistency.

To facilitate mutual adaptation between the template and search branches, we employ a bidirectional cross-attention mechanism. Each branch leverages contextual cues from the other for progressive refinement:

$$\begin{aligned}\widetilde{\mathbf{F}}_l^{(T)} = \text{Cross-Attention}\Big(&\mathbf{Q} = \mathbf{F}_{l-1}^{(T)}\mathbf{W}_Q^T,\\ &\mathbf{K} = \mathbf{F}_{l-1}^{(S)}\mathbf{W}_K^T, \mathbf{V} = \mathbf{F}_{l-1}^{(S)}\mathbf{W}_V^T\Big)\end{aligned} \tag{2}$$

$$\begin{aligned}\widetilde{\mathbf{F}}_l^{(S)} = \text{Cross-Attention}\Big(&\mathbf{Q} = \mathbf{F}_{l}^{(S)}\mathbf{W}_Q^S,\\ &\mathbf{K} = \mathbf{F}_{l-1}^{(T)}\mathbf{W}_K^S, \mathbf{V} = \mathbf{F}_{l-1}^{(T)}\mathbf{W}_V^S\Big)\end{aligned} \tag{3}$$

where $\mathbf{W}$ denotes the projection matrices. Here, the template branch uses its previous output as the query to attend to the search features, while the search branch uses a densely propagated representation as the query to capture updated cues from the template. This symmetrical interaction ensures bidirectional semantic alignment and cross-branch complementarity.

While cross-attention captures global context, it may dilute fine-grained spatial structures. To preserve local geometry, we refine features via *Local Self-Attention (LSA)* within each point's local neighborhood. This module integrates both feature and positional cues from k nearest neighbors in 3D space. Defining position-enhanced features as:

$$\begin{aligned}\widehat{\mathbf{F}}_l^{(T)} &= \widetilde{\mathbf{F}}_l^{(T)} + \phi\big(\mathbf{P}^{(T)}\big),\\ \widehat{\mathcal{N}}_k^{(T)} &= \mathcal{N}_k\big(\widetilde{\mathbf{F}}_l^{(T)}\big) + \phi\big(\mathcal{N}_k(\mathbf{P}^{(T)})\big)\end{aligned} \tag{4}$$

$$\begin{aligned}\widehat{\mathbf{F}}_l^{(S)} &= \widetilde{\mathbf{F}}_l^{(S)} + \phi\big(\mathbf{P}^{(S)}\big),\\ \widehat{\mathcal{N}}_k^{(S)} &= \mathcal{N}_k\big(\widetilde{\mathbf{F}}_l^{(S)}\big) + \phi\big(\mathcal{N}_k(\mathbf{P}^{(S)})\big)\end{aligned} \tag{5}$$

The local attention operations are then expressed as:

$$\begin{aligned}\mathbf{F}_l^{(T)} = \text{Self-Attention}\Big(&\mathbf{Q} = \widehat{\mathbf{F}}_l^{(T)}\mathbf{W}_Q^T,\\ &\mathbf{K} = \widehat{\mathcal{N}}_k^{(T)}\mathbf{W}_K^T, \mathbf{V} = \widehat{\mathcal{N}}_k^{(T)}\mathbf{W}_V^T\Big)\end{aligned} \tag{6}$$

$$\begin{aligned}\mathbf{F}_l^{(S)} = \text{Self-Attention}\Big(&\mathbf{Q} = \widehat{\mathbf{F}}_l^{(S)}\mathbf{W}_Q^S,\\ &\mathbf{K} = \widehat{\mathcal{N}}_k^{(S)}\mathbf{W}_K^S, \mathbf{V} = \widehat{\mathcal{N}}_k^{(S)}\mathbf{W}_V^S\Big)\end{aligned} \tag{7}$$

where $\phi(\cdot)$ denotes a learnable MLP applied to 3D coordinates for positional encoding. $\mathcal{N}_k(\cdot)$ represents the k-nearest neighbor operator based on Euclidean distances in 3D space. The query $\mathbf{Q}$ is derived from the center point, while keys and values are aggregated from its local neighbors. Multi-head attention is applied followed by residual MLP and normalization for refinement.

Since higher layers encode richer semantic relationships through progressive refinement, the final fused feature is taken from the output of the last layer.

3.4. Temporal Context Fusion

Relying solely on single-frame features is insufficient under occlusion or drastic appearance changes. Therefore, after spatial semantic modeling via DCP-net, we design a Temporal Context Fusion module to enhance temporal consistency by fusing historical frame features.

As shown in Fig. 1, the TCF module takes the current frame t's fused feature map $\mathbf{F}_\text{t} \in \mathbb{R}^{N_s \times C}$ (output of DCP-net) and corresponding 3D coordinates $\mathbf{P}_\text{t} \in \mathbb{R}^{N_s \times 3}$ as input. It utilizes a Memory Bank storing fused features $\{\mathbf{F}_{t-T}, ..., \mathbf{F}_{t-1}\}$ and their corresponding 3D coordinates $\{\mathbf{P}_{t-T}, ..., \mathbf{P}_{t-1}\}$ from the recent T frames. TCF first applies linear projections to generate Query ($\mathbf{Q}_\text{t}$), Key ($\mathbf{K}_\text{hist}^{(j)}$), and Value ($\mathbf{V}_\text{hist}^{(j)}$) vectors for the current frame and each historical frame $j \in \{t-T, ..., t-1\}$:

$$\mathbf{Q}_\text{t} = \mathbf{F}_\text{t}\mathbf{W}_Q, \quad \mathbf{K}_\text{hist}^{(j)} = \mathbf{F}_j\mathbf{W}_K, \quad \mathbf{V}_\text{hist}^{(j)} = \mathbf{F}_j\mathbf{W}_V \tag{8}$$

For each historical frame j, a channel-wise attention mechanism computes point-wise matching weights between the current frame and frame j. Specifically, for point i in the current frame and point m in frame j, the attention score considers feature similarity and spatial position difference:

$$e_{im}^{(j)} = \text{MLP}_\text{score}\left(\mathbf{q}_{\text{t},i} - \mathbf{k}_{\text{hist},m}^{(j)} + \phi(\mathbf{p}_{\text{t},i} - \mathbf{p}_{j,m})\right) \tag{9}$$

where $\mathbf{q}_{\text{t},i}$ is the query feature of current point i, $\mathbf{k}_{\text{hist},m}^{(j)}$ is the key feature of point m in frame j, $\mathbf{x}_{\text{t},i}$ and $\mathbf{x}_{j,m}$ are their 3D coordinates, ϕ is a positional encoding function (e.g., sinusoidal or MLP), and MLP_score consists of two linear layers and one nonlinear ReLU. Scores are normalized

per historical frame j via softmax:

$$\alpha_{im}^{(j)} = \frac{\exp(e_{im}^{(j)})}{\sum_{m'} \exp(e_{im'}^{(j)})} \tag{10}$$

The aggregated feature for current point i from frame j is:

$$\mathbf{f}_i^{(j)} = \sum_m \alpha_{im}^{(j)} \mathbf{v}_{\text{hist},m}^{(j)} \tag{11}$$

This yields aligned features $\mathbf{f}^{(j)} \in \mathbb{R}^{N_s \times C}$ for each history frame j.

To weight different historical frames, the aligned features $\{\mathbf{f}^{(t-T)}, \ldots, \mathbf{f}^{(t-1)}\}$ are stacked along a new dimension. The current frame features are projected as the query via $\mathbf{Q}_{\text{frame}} = \mathbf{F}_{\text{t}} \mathbf{W}_{Q_f}$, and each historical frame feature $\mathbf{f}^{(j)}$ is projected as the key via $\mathbf{K}_{\text{frame},j} = \mathbf{f}^{(j)} \mathbf{W}_{K_f}$, where $j = t-T, \ldots, t-1$. Then, the frame-level attention weights are computed as:

$$w_j = \text{MLP}_{\text{frame}} \left([\mathbf{Q}_{\text{frame}}; \mathbf{K}_{\text{frame},j}] \right), \tag{12}$$

$$\beta_j = \frac{\exp(w_j)}{\sum_{k=t-T}^{t-1} \exp(w_k)}, \quad j = t-T, \ldots, t-1 \tag{13}$$

The final temporally enhanced feature is obtained via weighted fusion and residual connection:

$$\mathbf{F}_{\text{temp}} = \mathbf{F}_{\text{t}} + \sum_{j=t-T}^{t-1} \beta_j \mathbf{f}^{(j)} \tag{14}$$

Here, $\text{MLP}_{\text{score}}$ and $\text{MLP}_{\text{frame}}$ are both two-layer perceptrons with ReLU activation, mapping C-dimensional inputs to scalar weights.

The TCF module can adaptively integrate semantic cues from history, enhancing temporal continuity and robustness. $\mathbf{F}_{\text{temp}}$ is used by the localization head to predict the target's 3D box.

3.5. Localization Head

To regress the 3D target state from temporally fused features, we follow STNet [11] and adopt the voxel-to-BEV localization framework introduced in V2B [10]. Point-wise features $\mathbf{F}_{\text{temp}} \in \mathbb{R}^{N_s \times C}$ are first voxelized into a regular grid and compressed along the Z-axis to form a BEV representation, which is then processed by a 2D convolutional decoder with three output branches: a heatmap head for center classification, an offset-rotation head for local refinement, and a Z-coordinate head for height prediction. Following [10], the total loss is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{shape}} + \lambda_2 \left(\mathcal{L}_{\text{center}} + \mathcal{L}_{\text{off}} \right) + \lambda_3 \mathcal{L}_z \tag{15}$$

where λ_1, λ_2, and λ_3 are hyperparameters balancing shape generation, 2D center and offset regression, and z-axis position regression, respectively. The detailed formulations of each loss component and the specific hyperparameter settings can be found in [10].

4. Experiments

4.1. Experimental Settings

4.1.1 Dataset

We conduct our experiments on two widely used benchmarks for 3D single object tracking: KITTI [7] and nuScenes [1], which together offer sufficient diversity to evaluate tracking performance across varied driving conditions. KITTI dataset provides front-view LiDAR and high-resolution camera data in urban driving scenarios, while nuScenes dataset offers full-surround 360° LiDAR coverage and richer annotations, including object attributes and trajectory information. Following the dataset splits used in previous studies [11, 23, 33], we divide the 21 training sequences of the KITTI dataset into training (sequences 0–16), validation (sequences 17–18), and test (sequences 19–20) sets. The nuScenes dataset contains 700 training and 150 validation sequences. Following [10, 11], we use the model trained on the KITTI and evaluate it on the nuScenes validation set to assess its generalization.

4.1.2 Implementation Details

We adopt STNet [11] as our baseline. We sample $N_t = 512$ template points and $N_s = 1024$ search points, which are fed into the model as input. The backbone adopts the design of STNet [11], while the regression head follows the design of V2B [10]. For the proposed DCP-net, we set the number of layers to $L = 2$ and the local self-attention (LSA) neighborhood size to $k = 48$. The cross-attention modules adopt the linear-complexity Transformer architecture from [13], with 2 attention heads. For the TCF module, we set the history length to $T = 3$. The final output feature has a dimension of 1024×32.

4.1.3 Evaluation Metrics

We use Success and Precision from One Pass Evaluation (OPE) [14]. Success measures the Area Under the Curve (AUC) of the 3D Intersection over Union (IoU) between predicted and ground-truth boxes. Precision measures the AUC of the distance threshold success plot for center distance errors within [0, 2] meters.

4.2. Results on KITTI

We evaluate DT-Tracker on the KITTI benchmark (Table 1), comparing it with recent methods across four categories: Car, Pedestrian, Van, and Cyclist, including [5, 11, 16, 21, 31, 32], etc. DT-Tracker achieves the highest Success rate on Car (72.7%) and Van (59.9%), and the second-best Success on Cyclist (74.4%). It also attains the highest Precision on Car (84.6%), Van (71.1%), and Cyclist (94.2%),

Table 1. Performance comparison on the KITTI and nuScenes datasets. Success/Precision are reported per category, and the mean is weighted by frame count. Best results are marked in **bold**.

Method	KITTI					nuScenes				
	Car 6424	Pedestrian 6088	Van 1248	Cyclist 308	Mean 14068	Car 15578	Pedestrian 8019	Truck 3710	Bicycle 501	Mean 27808
SC3D[8]	41.3/57.9	18.2/37.8	40.4/47.0	41.5/70.4	31.1/48.5	25.0/27.1	14.2/16.2	**25.7/21.9**	17.0/18.2	21.8/23.1
P2B[23]	56.2/72.8	28.7/49.6	40.8/48.4	32.1/44.7	42.4/60.0	27.0/29.2	15.9/22.0	21.5/16.2	20.0/26.4	22.9/25.3
LTTR[3]	65.0/77.1	33.2/56.8	35.8/45.6	66.2/89.9	48.7/65.8	–	–	–	–	–
PTT[24]	67.8/81.8	44.9/72.0	43.6/52.5	37.2/47.3	55.1/74.2	–	–	–	–	–
BAT[33]	60.5/77.7	42.1/70.1	52.4/67.0	33.7/45.4	50.0/69.9	22.5/24.1	17.3/24.5	19.3/15.8	17.0/18.8	20.5/23.0
V2B[10]	70.5/81.3	48.3/73.5	50.1/58.0	40.8/49.7	58.4/75.2	31.3/35.1	17.3/23.4	21.7/16.7	**22.2**/19.1	25.8/29.0
SMAT[4]	71.9/82.4	52.1/81.5	41.4/53.2	61.2/87.3	60.4/79.5	–	–	–	–	–
STNet[11]	72.1/84.0	49.9/77.2	58.0/70.6	73.5/93.7	61.3/80.1	32.2/36.1	19.1/27.2	22.3/16.8	21.2/29.2	26.9/30.8
MLSENet[27]	69.7/81.0	50.7/80.0	55.2/64.8	41.0/49.7	59.6/78.4	–	–	–	–	–
GLT-T[21]	68.2/82.1	52.4/78.8	52.6/62.9	68.9/92.1	60.1/79.3	–	–	–	–	–
STTacker[5]	66.5/79.9	**60.4/89.4**	50.5/63.6	**75.3**/93.9	62.6/**82.9**	–	–	–	–	–
CDTracker[31]	71.7/83.1	49.2/75.9	51.7/62.3	71.2/93.2	60.2/78.4	–	–	–	–	–
Li's method[16]	70.3/82.0	57.1/83.9	48.4/56.9	73.7/94.0	62.7/80.9	–	–	–	–	–
OST[32]	72.0/84.2	51.4/82.6	57.5/68.2	49.2/60.4	61.3/81.6	26.6/28.0	14.8/16.2	19.3/14.7	17.0/16.4	22.1/22.6
DT-Tracker(Ours)	**72.7/84.6**	52.6/78.5	**59.9/71.1**	74.4/**94.2**	**62.8**/80.9	**32.6/36.7**	**19.6/27.4**	22.0/16.6	21.7/**30.1**	**27.2/31.2**

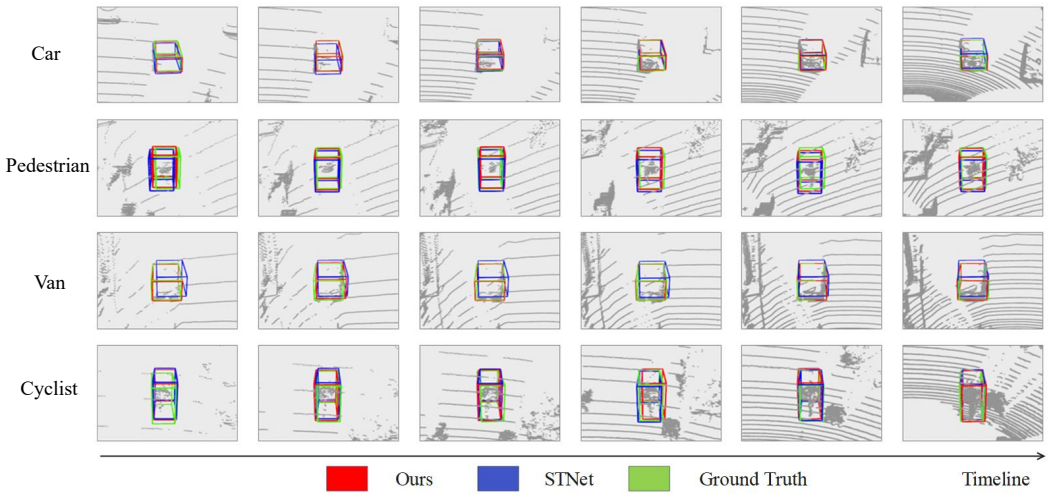

Figure 2. Visualization of tracking results on KITTI dataset. We compare our DT-Tracker with STNet[11], and show the cases of Car, Pedestrian, Van and Cyclist categories.

indicating strong localization accuracy across these categories. For the Pedestrian category, DT-Tracker achieves a Success of 52.6% and Precision of 78.5%, showing clear improvements over the baseline STNet [11], but there is still room for enhancement in this category, suggesting that future enhancements may focus on better handling small, deformable, or highly occluded pedestrian targets. In terms of overall performance, DT-Tracker achieves the highest mean Success (62.8%) and a strong mean Precision (80.9%), surpassing most compared methods. Compared to the baseline STNet [11], DT-Tracker improves Success by +0.6% on Car and shows consistent gains across all categories. For visualization, we report the comparison results between our method and baseline on the KITTI dataset in Fig. 2, covering Car, Pedestrian, Van and Cyclist categories. Furthermore, despite both utilizing temporal context, DT-Tracker outperforms STTracker [5] in three out of four categories and achieves a higher overall Success, demonstrating the effectiveness of our dual-context design and temporal fusion strategy.

4.3. Results on nuScenes

We evaluate DT-Tracker on the more challenging nuScenes dataset, which features a lower annotation frequency (2 Hz vs. KITTI's 10 Hz) and more diverse, complex scenes. Our model, trained solely on KITTI, is compared with several representative methods, including [8, 10, 23, 33], to assess its cross-dataset generalization. As shown in Ta-

ble 1, DT-Tracker achieves the highest Success on Car (32.6%) and Pedestrian (19.6%), and the highest Precision on Car (36.7%), Pedestrian (27.4%), and Bicycle (30.1%). In terms of overall performance, DT-Tracker obtains the highest mean Success/Precision of 27.2% / 31.2%. Compared with the baseline STNet [11], DT-Tracker improves mean Success from 26.9% to 27.2%, and mean Precision from 30.8% to 31.2%, demonstrating stronger generalization despite no access to nuScenes data during training. The relatively modest gains may be attributed to both the lower annotation frequency and the greater scene complexity in nuScenes.

4.4. Running speed

We evaluate runtime on the KITTI car category using a single RTX 4090 GPU. DT-Tracker achieves 17 FPS, including 7.3ms for data processing, 48.2ms for network forward propagation, and 3.4ms for post-processing, and STNet [11] reaches 20 FPS under the same platform. Despite the added dual-context and temporal fusion computations, our tracker remains real time with improved robustness.

Table 2. Ablation study of the impact of DCP-net and TCF modules on Car and Van categories.

DCP-net	TCF module	Car	Van
×	×	72.1/84.0	58.0/70.6
✓	×	72.4/84.2	59.1/70.8
×	✓	72.6/84.4	59.5/71.0
✓	✓	72.7/84.6	59.9/71.1

Table 3. Comparison of different layer numbers (L) and attention directions in DCP-net on Car and Van categories.

L	Bi-direction		Single	
	Car	Van	Car	Van
1	71.8/83.7	58.1/70.3	71.2/83.3	57.4/69.3
2	72.7/84.6	59.9/71.1	71.8/83.8	59.3/70.7
3	72.4/84.2	59.2/70.8	71.5/83.6	58.8/70.0

4.5. Ablation Studies

4.5.1 Component-wise Contributions

We conduct ablation experiments on the Car and Van categories of the KITTI dataset to evaluate the contributions of the Dual-Context Propagation Network (DCP-net) and the Temporal Context Fusion (TCF) module. As shown in Table 2, using only DCP-net yields 72.4%/84.2% (Success/Precision) on Car and 59.1%/70.8% on Van. The standalone TCF module achieves 72.6%/84.4% on Car and 59.5%/71.0% on Van. When combining both components, DT-Tracker achieves the best results: 72.7%/84.6% on Car and 59.9%/71.1% on Van. These results demonstrate that DCP-net and TCF are complementary and both contribute to the overall performance.

Table 4. Impact of varying the number of historical frames (T) in the TCF module on Car and Van categories.

T	Car	Van
2	72.2/84.2	58.7/70.7
3	72.7/84.6	59.9/71.1
4	72.3/84.3	59.2/70.9

4.5.2 Impact of Network Depth

We further evaluate the effect of DCP-net configurations by varying the number of layers (L) and comparing bidirectional versus unidirectional attention (Table 3). Across all settings, bidirectional attention consistently outperforms its single-direction counterpart, confirming the benefit of symmetric feature interaction. The optimal configuration is $L = 2$ layers with bidirectional attention, achieving 72.7%/84.6% on Car and 59.9%/71.1% on Van. Increasing to $L = 3$ leads to a slight drop in performance, possibly due to feature over-smoothing or redundant interactions.

4.5.3 Temporal History Length

We study the effect of the number of historical frames T used in the TCF module. As shown in Table 4, using $T = 3$ achieves the best performance on both Car (72.7%/84.6%) and Van (59.9%/71.1%). Reducing to $T = 2$ lowers the results slightly, while increasing to $T = 4$ does not yield further improvement. These results indicate that three history frames provide a good balance between temporal context and computational efficiency.

5. Conclusion

In this paper, we proposed DT-Tracker, a 3D object tracking framework that integrates Dual-Context Propagation and Temporal Context Fusion for robust spatiotemporal modeling. Experiments on KITTI and nuScenes demonstrate strong accuracy and cross-dataset generalization. While effective, DT-Tracker still lacks a mechanism to evaluate the quality of historical frames. Future work will explore adaptive frame selection to retain informative frames and discard redundant ones, improving both robustness and efficiency.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Gi-

ancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020. 2, 6

[2] Andrew I. Comport, Éric Marchand, and François Chaumette. Robust model-based tracking for robot vision. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28 - October 2, 2004*, pages 692–697. IEEE, 2004. 1

[3] Yubo Cui, Zheng Fang, Jiayao Shan, Zuoxu Gu, and Sifan Zhou. 3d object tracking with transformer. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 317. BMVA Press, 2021. 1, 2, 3, 7

[4] Yubo Cui, Jiayao Shan, Zuoxu Gu, Zhiheng Li, and Zheng Fang. Exploiting more information in sparse point cloud for 3d single object tracking. *IEEE Robotics Autom. Lett.*, 7(4): 11926–11933, 2022. 7

[5] Yubo Cui, Zhiheng Li, and Zheng Fang. Sttracker: Spatio-temporal tracker for 3d single object tracking. *IEEE Robotics Autom. Lett.*, 8(8):4967–4974, 2023. 1, 3, 6, 7

[6] Shihao Feng, Pengpeng Liang, Jin Gao, and Erkang Cheng. Multi-correlation siamese transformer network with dense connection for 3d single object tracking. *IEEE Robotics Autom. Lett.*, 8(12):8066–8073, 2023. 2, 3, 5

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3354–3361. IEEE Computer Society, 2012. 2, 6

[8] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1359–1368. Computer Vision Foundation / IEEE, 2019. 1, 2, 7

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. 5

[10] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3d siamese voxel-to-bev tracker for sparse point clouds. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28714–28727, 2021. 1, 2, 6, 7

[11] 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 *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part II*, pages 293–310. Springer, 2022. 2, 3, 4, 6, 7, 8

[12] Haobo Jiang, Kaihao Lan, Le Hui, Guangyu Li, Jin Xie, Shangbing Gao, and Jian Yang. Point cloud registration-driven robust feature matching for 3-d siamese object tracking. *IEEE Trans. Neural Networks Learn. Syst.*, 36(1):967–977, 2025. 3

[13] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 5156–5165. PMLR, 2020. 6

[14] Matej Kristan, Jiri Matas, Ales Leonardis, Tomás Vojír, Roman P. Pflugfelder, Gustavo Fernández, Georg Nebehay, Fatih Porikli, and Luka Cehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(11):2137–2155, 2016. 6

[15] Kuan-Hui Lee and Jenq-Neng Hwang. On-road pedestrian tracking across multiple driving recorders. *IEEE Trans. Multim.*, 17(9):1429–1438, 2015. 1

[16] Jian Li, Qi Wu, Chun Yi, Ke Chen, and Shiyu Xuan. 3d single object tracking network based on point cloud presegmentation. In *Proceedings of the 2024 43rd Chinese Control Conference (CCC)*, pages 8722–8727. IEEE, 2024. 2, 6, 7

[17] Jiaming Liu, Yue Wu, Maoguo Gong, Qiguang Miao, Wenping Ma, and Fei Xie. Instance-guided point cloud single object tracking with inception transformer. *IEEE Trans. Instrum. Meas.*, 72:1–12, 2023. 2, 3

[18] Zhe Liu, Chuanzhe Suo, Yingtian Liu, Yueling Shen, Zhijian Qiao, Huanshu Wei, Shunbo Zhou, Haoang Li, Xinwu Liang, Hesheng Wang, and Yun-Hui Liu. Deep learning-based localization and perception systems: Approaches for autonomous cargo transportation vehicles in large-scale, semi-closed environments. *IEEE Robotics Autom. Mag.*, 27(2): 139–150, 2020. 1

[19] Pawan Kumar Mishra and G. P. Saroha. A study on video surveillance system for object detection and tracking. In *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom '16)*, pages 221–226. IEEE, 2016. 1

[20] Jiahao Nie, Zhiwei He, Yuxiang Yang, Zhengyi Bao, Mingyu Gao, and Jing Zhang. OSP2B: one-stage point-to-box network for 3d siamese tracking. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 1285–1293. ijcai.org, 2023. 2, 3

[21] 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. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 1957–1965. AAAI Press, 2023. 1, 2, 3, 6, 7

[22] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), Octo-*

ber 27 - November 2, 2019, pages 9276–9285. IEEE, 2019. 1, 2

[23] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2B: point-to-box network for 3d object tracking in point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6328–6337. Computer Vision Foundation / IEEE, 2020. 1, 2, 3, 6, 7

[24] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. PTT: point-track-transformer module for 3d single object tracking in point clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 1310–1316. IEEE, 2021. 1, 2, 3, 7

[25] Shengjing Tian, Xiuping Liu, Meng Liu, Yuhao Bian, Junbin Gao, and Baocai Yin. Learning the incremental warp for 3d vehicle tracking in lidar point clouds. *Remote. Sens.*, 13(14): 2770, 2021. 2

[26] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *IEEE Trans. Vis. Comput. Graph.*, 16(3):355–368, 2010. 1

[27] Qiaoyun Wu, Changyin Sun, and Jun Wang. Multi-level structure-enhanced network for 3d single object tracking in sparse point clouds. *IEEE Robotics Autom. Lett.*, 8(1):9–16, 2023. 7

[28] Yan Xia, Qiangqiang Wu, Wei Li, Antoni B. Chan, and Uwe Stilla. A lightweight and detector-free 3d single object tracker on point clouds. *IEEE Trans. Intell. Transp. Syst.*, 24 (5):5543–5554, 2023. 1, 2

[29] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Cxtrack: Improving 3d point cloud tracking with contextual information. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 1084–1093. IEEE, 2023. 1, 3

[30] Yongchang Zhang, Hanbing Niu, Yue Guo, and Wenhao He. 3d single-object tracking with spatial-temporal data association. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, pages 264–269. IEEE, 2022. 1, 3

[31] Yuan Zhang, Chenghan Pu, Yu Qi, Jianping Yang, Xiang Wu, Muyuan Niu, and Mingqiang Wei. Cdtracker: Coarse-to-fine feature matching and point densification for 3d single-object tracking. *Remote. Sens.*, 16(13):2322, 2024. 6, 7

[32] Xiantong Zhao, Yinan Han, Shengjing Tian, Jian Liu, and Xiuping Liu. OST: efficient one-stream network for 3d single object tracking in point clouds. *IEEE Trans. Multim.*, 27: 990–1002, 2025. 2, 3, 6, 7

[33] 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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 13179–13188. IEEE, 2021. 1, 2, 3, 6, 7

[34] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8101–8110. IEEE, 2022. 1, 2

[35] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8521–8530. IEEE, 2022. 2, 3