IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Beyond Frame-wise Tracking: A Trajectory-based Paradigm for Efficient Point Cloud Tracking

BaiChen Fan[1†], Yuanxi Cui[2†], Jian Li[1], Qin Wang[1], Shibo Zhao[3], Muqing Cao[3], Sifan Zhou[3*‡]

***Abstract*—LiDAR-based 3D single object tracking (3D SOT) is a critical task in robotics and autonomous systems. Existing methods typically follow frame-wise motion estimation or a sequence-based paradigm. However, the two-frame methods are efficient but lack long-term temporal context, making them vulnerable in sparse or occluded scenes, while sequence-based methods that process multiple point clouds gain robustness at a significant computational cost. To resolve this dilemma, we propose a novel trajectory-based paradigm and its instantiation, TrajTrack. TrajTrack is a lightweight framework that enhances a base two-frame tracker by implicitly learning motion continuity from historical bounding box trajectories alone—without requiring additional, costly point cloud inputs. It first generates a fast, explicit motion proposal and then uses an implicit motion modeling module to predict the future trajectory, which in turn refines and corrects the initial proposal. Extensive experiments on the large-scale NuScenes benchmark show that TrajTrack achieves new state-of-the-art performance, dramatically improving tracking precision by 3.02% over a strong baseline while running at 55 FPS. Besides, we also demonstrate the strong generalizability of TrajTrack across different base trackers. Code is available at https://github.com/FiBonaCci225/TrajTrack.**

I. INTRODUCTION

3D single object tracking (3D SOT) is a fundamental task in computer vision with wide applications in autonomous driving and robotics. It aims to accurately track a target across frames using LiDAR or cameras. Unlike Multi-object tracking (MOT) [1]–[4], which is mostly based on the tracking-by-detection paradigm and highly sensitive to missed detections or identity switching, SOT only requires initial target annotation and can perform end-to-end tracking of the target without a detector, providing higher robustness and making it particularly suitable for robotic scenarios that require continuous tracking of a specified target. Compared to camera images, LiDAR point clouds offer notable advantages: robustness to light changes and the ability to capture rich geometries from environments. These strengths make it well-suited for robust perception in robotics [5], [6]. However, challenges from point clouds' inherent sparsity, especially under occlusion, and appearance changes due to motion or blur often lead to incomplete observations, making reliable tracking difficult.

Previous methods typically follow a two-frame paradigm (Fig.1(a)), which can be categorized into appearance-based [7]–[14] and motion-based approaches [15]–[20]. Similarity-based methods such as SC3D [7], P2B [9], and PTT [12] rely on frame-wise appearance cues between template and search point cloud and perform similarity matching, but they are sensitive to appearance changes and fast motion. In contrast, motion-based methods like M^2-Track series [15], [16] and P2P [21] explicitly model inter-frame motion to estimate relative translation. However, by relying solely on two-frame information, they all struggle in sparse or occluded scenes due to limited appearance or motion clues. More critically, they lack considerations of long-term motion continuity, preventing them from forming a predictive motion prior—for example, anticipating a vehicle's position as it moves through a temporary occlusion—which is essential for robust tracking.

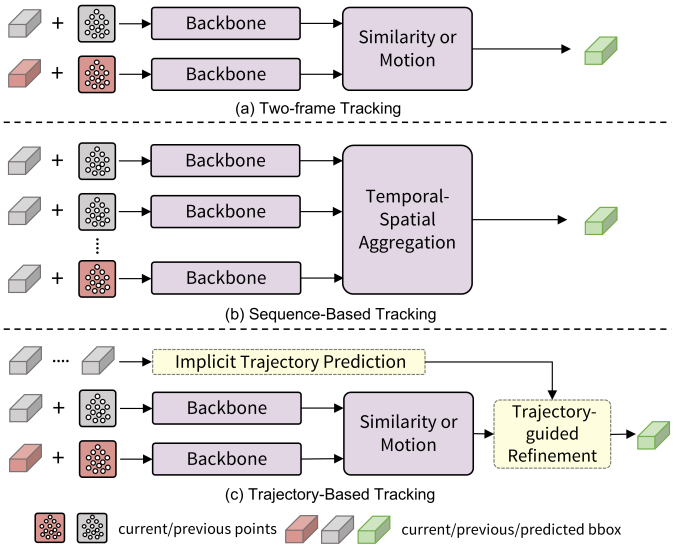

Fig. 1: **Different tracking paradigms.** (a) Two-frame paradigm exploits two-frame inputs for tracking through appearance matching or motion prediction. (b) Sequence-based paradigm uses multi-frames inputs to integrate target information. (c) Our Trajectory-based paradigm considers both short- and long-term motion clues.

To overcome these challenges, recent studies have explored the sequence-based paradigm (Fig.1(b)), which incorporates long-term temporal information by processing multiple point cloud frames. For instance, STTracker [22] processes a sequence of BEV feature maps and uses a deformable attention mechanism to aggregate spatio-temporal features. Going a step further, SeqTrack3D [23] utilizes sequences of both point clouds and their corresponding historical bounding boxes, using a Seq2Seq model to explicitly learn motion patterns. While these approaches improve robustness in sparse scenes, they introduce a critical trade-off. Their reliance on processing multi-frame point clouds inevitably leads to a high computational cost, making them less suitable for latency-sensitive, real-time tracking applications. Besides, their complex feature extraction from sequential data can still struggle to learn a

*Corresponding Author. †Equal Contribution. ‡Project Leader.
[1]BaiChen Fan, Jian Li, and Qin Wang are with the Institute of Quantum Information and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China.
[2]Yuanxi Cui is with the Shanghai Jiao Tong University, Shanghai, China.
[3]Sifan Zhou, Shibo Zhao and Muqing Cao are with the Robotics Institute, Carnegie Mellon University, PA, USA.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

clear, consistent motion trajectory, especially with noisy or occluded frames. This leaves a clear opening for a method that can leverage long-term motion continuity in a more lightweight and direct manner.

In this paper, we propose **TrajTrack**, a novel 3D SOT framework (Fig.1(c)), that synergistically combines the strengths of short-term explicit motion and long-term implicit continuity through a two-stage process. **Stage 1: Explicit Motion Proposal:** We first employ an efficient, two-frame explicit motion model. By analyzing two inter-frame point clouds, this stage rapidly generates an initial tracking proposal. This proposal effectively captures instantaneous motion but can be prone to errors in sparse or occluded scenes. **Stage 2: Implicit Trajectory Prediction:** This is the core innovation of our framework. We introduce an implicit motion modeling module that operates solely on the historical sequence of bounding box. It uses a lightweight Transformer to learn the object's long-term motion continuity and predict future trajectory with confidence. **Post-process: Trajectory-guided Proposal Refinement:** Finally, we design a proposal refinement mechanism. It leverages the predicted trajectory from Stage 2, which embodies a long-term motion prior, to intelligently calibrate and correct the potentially inaccurate initial proposal from Stage 1. By integrating short-term explicit observations with long-term, consistent motion patterns, TrajTrack effectively captures both local and global motion cues. Its key innovation lies in enhancing tracking robustness by exploiting the intrinsic continuity of object motion, without the high computational cost of processing multiple point cloud frames. Extensive experiments demonstrate the superior performance and real-time speed of our approach. In summary, our main contributions are as follows:

- **Trajectory-based Paradigm**: We propose a novel trajectory-based paradigm that leverages historical bboxes to incorporate long-term motion continuity, enhancing robustness without the overhead of multi-frame input.
- **TrajTrack with Implicit Motion Modeling**: We instantiate this paradigm in TrajTrack, a framework featuring a novel Implicit Motion Modeling (IMM) module to provide predictive priors to synergize long-term continuity with short-term observations.
- **SOTA Performance**: We achieve new state-of-the-art performance on the large-scale nuScenes benchmark by a significant margin (+4.48% in Precision). Besides, we demonstrate the strong generalizability by consistently improving the existing method's performance.

II. Related Work

Object Tracking on Point Clouds. The field of 3D single-object tracking (SOT) has advanced rapidly with the development of point cloud processing. The pioneering work SC3D [7] applied Siamese networks and template matching directly to point clouds, but suffered from inefficiency and non-end-to-end training. To address this, P2B [9] and 3D-SiamRPN [10] employed RPN and VoteNet [24] for efficient proposal generation. Subsequent methods enhanced feature representation: BAT [11] incorporated box-aware structural features, PTT [12], [13] leveraged transformers, and V2B [25] converted sparse points to dense BEV maps. Attention mechanisms were further integrated by STNet [26], LTTR [27], and PTTR [14] to propagate target-specific features. Appearance-based SOTs remain vulnerable under sparsity and occlusion, motivating motion-centric designs. M^2-Track [15] modeled inter-frame motion via a segmentation-and-tracking pipeline, extended to semi-supervised settings by M²Track++ [16]. DMT [28] leveraged historical bounding boxes for center prediction, while P2P [21] inferred relative motion directly from cropped point clouds of consecutive frames. However, these methods primarily rely on short-term temporal inputs, neglecting long-term history. To address this, TAT [29] aggregated historical templates via RNNs but struggled with low-quality inputs. STTracker [22] employed deformable attention to integrate spatio-temporal features from BEV sequences, and SeqTrack3D [23] proposed a Sequence-to-Sequence paradigm to capture both geometry and motion from point cloud sequences. While sequence-based approaches improve robustness, they struggle to balance accuracy and efficiency due to the high cost of multi-frame processing and the complexity of modeling consistent motion.

Similarly, 3D MOT exploits temporal cues but along different lines. Filter-based methods (AB3DMOT [1], SimpleTrack [2]) use Kalman or constant-velocity models implicitly maintain motion through state vectors but struggle with nonlinearity. Query-based tracking-by-attention methods (MUTR3D [3], ADA-Track [4]), propagate object identities across frames via attention yet remain detection-driven and focus on association rather than explicit motion modeling. Crucially, MOT methods reinitialize states from detector outputs each frame; by contrast, SOT bypasses per-frame detection through initial target specification, enabling detector-free, persistent, and end-to-end tracking.

Sequence Modeling for Motion. Learning motion patterns from temporal sequences provides an effective way to model long-term context. In trajectory prediction, VectorNet [30] and LaneRCNN [31] introduced vectorized and rasterized scene representations with graph neural networks to capture dynamic interactions. GATraj [32] further enhanced this through graph attention. TNT [33] and DenseTNT [34] improved long-term reasoning via hierarchical goal selection and dense trajectory generation. Generative approaches such as VAEs [35], CVAEs [36], and GANs [37] learned trajectory distributions in a probabilistic manner. Most of these methods adopt a sequence-to-sequence paradigm, predicting future motion from past trajectory features. Recently, Transformer-based architectures have become dominant due to their ability to model complex temporal dependencies, as demonstrated by S2TNet [38] for spatio-temporal interactions and AgentFormer [39] for agent-centric motion coherence, with CASPFormer [40] introducing causal spatial priors for socially compliant forecasting. The success of these models in learning rich motion patterns from sparse positional sequences directly informs our approach. We are inspired from this powerful sequence modeling paradigm for the specific task of 3D SOT, designing a lightweight IMM module to learn motion continuity from an object's past trajectory.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

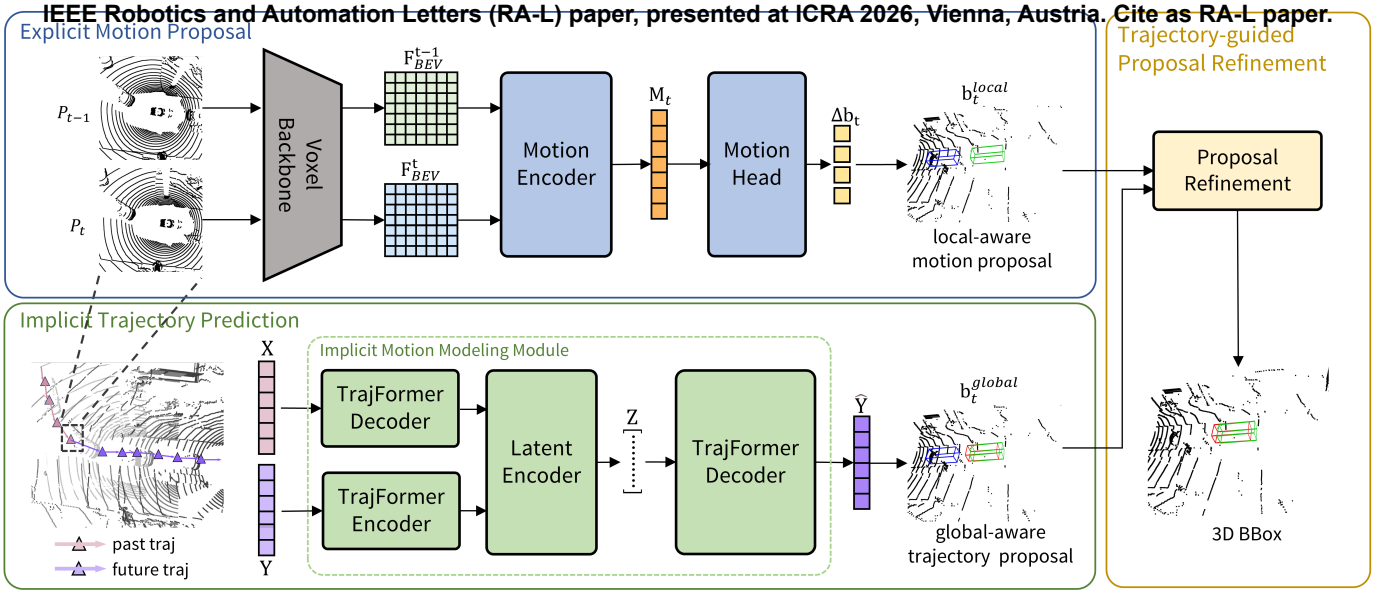

Fig. 2: Overview of the proposed **Trajectory-Based Paradigm** and its instantiation **TrajTrack**. Explicit Motion Proposal uses a two-frame tracking baseline to obtain the local-aware motion proposal. Implicit Trajectory Prediction is used to learn the object's global-aware motion continuity for trajectory proposal. Finally, Trajectory-guided Proposal Refinement cooperates the local and global-aware motion cues to get the refined 3D BBOX as the final output.

III. Method

A. Preliminaries

3D Single Object Tracking. In the 3D SOT task, given a sequence of point clouds $\mathbf{P}_t \in \mathbb{R}^{N\times 3}$ at time t with N points and the initial 3D bounding box $\mathbf{b}_t = (x_t, y_t, z_t, h_t, w_t, l_t, \theta_t) \in \mathbb{R}^{1\times 7}$ in the initial frame t, where (x,y,z) and (h,w,l) represent the center coordinate and size and θ represent the rotation angle. The goal of 3D SOT aims to predict the box $\mathbf{b}_s = (x_s, y_s, z_s, h_s, w_s, l_s, \theta_s) \in \mathbb{R}^{1\times 7}$ in subsequent frames from the search area point cloud. Generally, the object size (h,w,l) will not change in all frames; therefore, the position parameters $(x_s, y_s, z_s, \theta_s)$ need to be predicted.

B. Framework Overview

Our **TrajTrack** framework is designed to address the challenges of 3D SOT by synergistically combining short-term and long-term motion cues. As illustrated in Fig. 2, it follows a two-stage "propose-predict-refine" pipeline: **(1) Explicit Motion Proposal:** First, a two-frame-based explicit motion model rapidly generates an initial, locally-aware tracking proposal for the current frame. **(2) Implicit Trajectory Prediction:** Second, our novel Implicit Motion Modeling (IMM) module predicts a globally-aware future trajectory. **(3) Trajectory-Guided Proposal Refinement:** Finally, a post-processing operation from the refinement mechanism uses the trajectory prior to intelligently correct the initial proposal, outputting a more robust tracking result.

C. Stage 1: Explicit Motion Proposal

This stage aims to rapidly generate a high-quality initial proposal that captures instantaneous motion. Recent works [21], [41] have demonstrated the superiority of voxels as a 3D representation in single object tracking (SOT). Therefore, we employ an efficient, voxel-based backbone from [21] to extract BEV features $\mathbf{F}_{t-1}, \mathbf{F}_t \in \mathbb{R}^{H\times W}$ from consecutive point clouds $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$. Notably, we can also use other 3D Tracker [9], [15] in this stage; we also provide the details in Tab. III. These feature maps are concatenated and passed through a motion encoder and a motion head to predict the inter-frame relative motion $\Delta\mathbf{b}_t$:

$$\mathbf{M}t = \mathrm{ME}([\mathbf{F}^{t-1}BEV, \mathbf{F}^{t}_{BEV}]) \quad (1)$$

$$\Delta\mathbf{b}_t = \mathrm{MH}(\mathbf{M}t) \quad (2)$$

where ME and MH are motion encoder and motion head, respectively. Finally, the initial motion proposal $\mathbf{b}_t^{\mathrm{local}}$ for the current frame is obtained by applying this displacement to the previous frame's bounding box $\mathbf{b}t-1$:

$$\mathbf{b}_t^{\mathrm{local}} = \mathbf{b}_{t-1} + \Delta\mathbf{b}_t \quad (3)$$

This local-aware proposal effectively captures local motion but can be prone to errors in sparse or occluded scenes. More details about the Motion Encoder and head can refer to [21].

D. Stage 2: Implicit Trajectory Prediction

The core of our framework is the Implicit Motion Modeling (IMM) module, which learns long-term motion continuity to predict a robust future trajectory. Critically, this module operates solely on the lightweight historical sequence of past bounding box coordinates, $\mathbf{X} = (x^{-H+1}, x^{-H+2}, \ldots, x^0)$, avoiding the high cost of processing multiple point clouds. The IMM module is trained to predict the future trajectory $\mathbf{Y} = (y^0, y^1, \ldots, y^T)$ conditioned on the past trajectory $\mathbf{X}$ and a latent variable $\mathbf{Z}$ that captures the inherent stochasticity of motion. H and T denote historical/future timesteps, respectively. We utilize the input information to predict a set of future trajectories $\hat{\mathbf{Y}}$, from which the trajectory position at the corresponding time step is selected as the global-aware trajectory proposal $\mathbf{b}_t^{global}$. The IMM consists of two main components, which are both implemented using our proposed TrajFormer architecture (Fig. 3).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

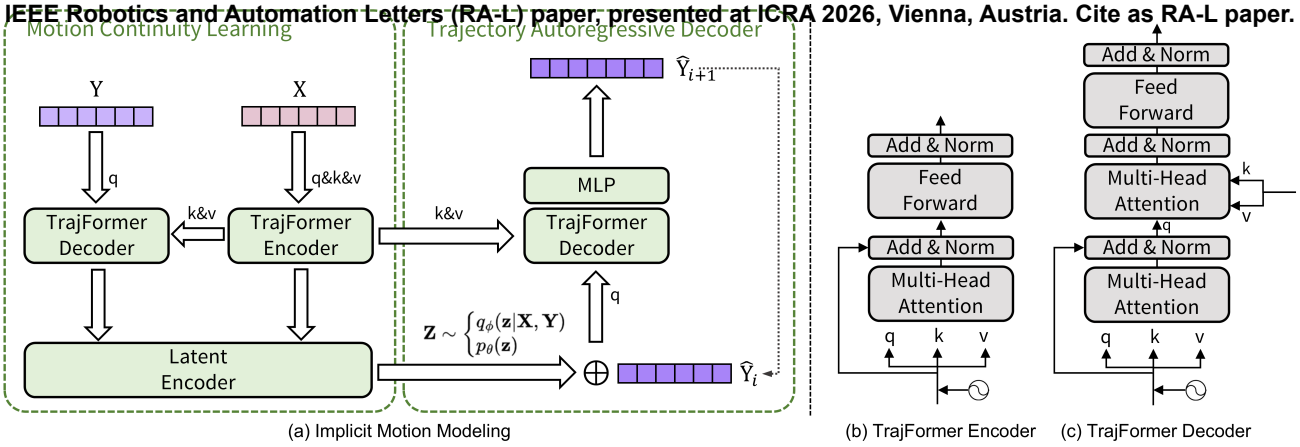

Fig. 3: **(a) Framework of Implicit Trajectory Prediction. (b) The TrajFormer Encoder. (c) The TrajFormer Decoder.**

Motion Continuity Learning. This component learns a latent representation of motion dynamics. We derive motion-centric features from the observed trajectory $\mathbf{X}$ and encode them with TrajFormer encoder and use a MLP to predict statistical parameters(μ, σ) to obtain prior distribution $p_\theta(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mu^p, \mathrm{Diag}(\sigma^p)^2)$. During training, the TrajFormer decoder integrates the future trajectory $\mathbf{Y}$ as conditional input and fuses it with the observed history to predict the posterior distribution $q_\phi(\mathbf{Z}|\mathbf{Y},\mathbf{X}) = \mathcal{N}(\mu^q, \mathrm{Diag}(\sigma^q)^2)$. A KL divergence loss enforces consistency between the prior and posterior, enabling the latent variable to capture expressive motion patterns.

Trajectory Autoregressive Decoder. The prediction process is initiated by sampling a latent code $\mathbf{z}$ rom the appropriate distribution (the prior p_θ at inference, the posterior q_ϕ at training). This code $\mathbf{z}$, which encapsulates the global motion intent, is concatenated with the last observed state embedding $\mathbf{x}^0$ to form an initial feature vector $\mathbf{f}^0 = \mathbf{x}^0 \oplus \mathbf{Z}$. This initial vector initiates the Trajectory Autoregressive Decoder (TAD) through recurrent updates:

$$\mathbf{f}_t = \mathbf{y}_t \oplus \mathbf{Z}, \quad \mathbf{Z} \sim \begin{cases} q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{Y}) & \textit{(training)} \\ p_\theta(\mathbf{Z}|\mathbf{X}) & \textit{(inference)} \end{cases} \tag{4}$$

where $\mathbf{F}_\mathbf{Y}^t = \{\mathbf{f}^0, ..., \mathbf{f}^t\}$ represents the progressively augmented feature sequence. The decoding mechanism maintains persistent latent guidance by keeping the sampled $\mathbf{Z}$ constant throughout generation, ensuring global motion pattern consistency. Simultaneously, it accumulates dynamic temporal context through the expanding feature sequence $\mathbf{F}_t$, which explicitly preserves historical state dependencies. At each timestep t, the decoder predicts the next trajectory feature $\mathbf{F}_{t+1}$ conditioned on this dual-information representation and then transformed through an MLP to reconstruct the predicted trajectory $\mathbf{y}_{t+1}$. Finally, we get the predicted future trajectory $\hat{\mathbf{Y}}$ of length t_f. The predicted position for the current timestep serves as our final global-aware trajectory proposal, $\mathbf{b}_t^{global}$.

TrajFormer Architecture. Both the encoder and decoder are built upon our TrajFormer block, which adapts the standard Transformer architecture for trajectory modeling. To provide temporal context, we first add a cosine-based positional encoding scheme to the input trajectory coordinates. A TrajFormer block consists of two main sub-layers: multi-head self-attention (MHSA) and a feed-forward network (FFN). For an input sequence $\mathbf{X}_{in}$, the $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ matrices are first projected:

$$\mathbf{Q} = (\mathbf{X} + \mathrm{PE}(t_p))\mathbf{W}_Q, \tag{5}$$

$$\mathbf{K} = (\mathbf{X} + \mathrm{PE}(t_p))\mathbf{W}_K, \tag{6}$$

$$\mathbf{V} = (\mathbf{X} + \mathrm{PE}(t_p))\mathbf{W}_V \tag{7}$$

The self-attention output is then calculated, followed by residual connections, layer normalization, and an FFN block:

$$\mathbf{X}_{\text{attn}} = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{8}$$

$$\mathbf{X}' = \mathrm{LayerNorm}(\mathbf{X}_{in} + \mathbf{X}_{\text{attn}}), \tag{9}$$

$$\mathbf{X}_{\text{out}} = \mathrm{LayerNorm}(\mathbf{X}' + \mathrm{FFN}(\mathbf{X}')) \tag{10}$$

The TrajFormer Decoder block is similar but includes an additional cross-attention layer to incorporate the encoded features from the past trajectory.

Insight and Advantages of IMM. The core insight behind the IMM is the decoupling of long-term motion modeling from the high-bandwidth point cloud data. Unlike sequence-based methods that are burdened by processing multiple, dense point clouds to extract motion cues, our IMM operates on a highly compressed, low-dimensional representation: the historical trajectory of bounding box centers. This design is motivated by the observation that for tracking, the object's macro-level motion continuity (where it's going) is often more critical for overcoming occlusions and sparsity than its micro-level surface details in every historical frame. By learning from the trajectory alone, the IMM can focus exclusively on capturing the object's dynamic behavior over time—such as velocity and turning patterns—creating a robust, global motion prior. This lightweight, information-centric approach is the key to how **TrajTrack** achieves the robustness of a sequence-based method while only introducing a minor computational burden compared to a two-frame tracker.

E. Trajectory-guided Proposal Refinement

The final post-process of TrajTrack is to dynamically fuse the proposals from the explicit and implicit motion models. This stage receives two inputs: the agile but potentially noisy local-aware proposal, $\mathbf{b}_t^{\text{local}}$, and the stable, global-aware trajectory proposal, $\mathbf{b}_t^{\text{global}}$. To synergize these two, we introduce a confidence-based refinement strategy. We use the Intersection-over-Union (IoU) between the two proposals

IEEE Robotics and Automation Letters (RAL) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Comparison with SOTA methods on the nuScenes dataset. Success and Precision are used for evaluation. **Bold** and underline denote the best result and the second-best one respectively. * 64159 indicates the number of instances of cars

Method	Publish	Paradigm	Car [64,159]*	Pedestrian [33,227]	Truck [13,587]	Trailer [3,352]	Bus [2,953]	Mean [117,278]
SC3D [7]	CVPR'19	Two-frame	22.31 / 21.93	11.29 / 12.65	35.28 / 28.12	35.28 / 28.12	29.35 / 24.08	20.70 / 20.20
P2B [9]	CVPR'20		38.81 / 43.81	28.39 / 52.24	48.96 / 40.05	48.96 / 40.05	32.95 / 27.41	36.48 / 45.08
PTT [13]	IROS'21		41.22 / 45.26	19.33 / 32.03	50.23 / 48.56	51.70 / 46.50	39.40 / 36.70	36.33 / 41.72
BAT [11]	ICCV'21		40.73 / 43.29	28.83 / 53.32	52.59 / 44.89	52.59 / 44.89	35.44 / 28.01	38.10 / 45.71
V2B [25]	NIPS'21		54.40 / 59.70	30.10 / 55.40	53.70 / 54.50	54.90 / 51.44	- / -	- / -
M^2-Track [15]	CVPR'22		55.85 / 65.09	32.10 / 60.92	57.36 / 59.54	57.61 / 58.26	51.39 / 51.44	49.23 / 62.73
PTTR [14]	CVPR'22		51.89 / 58.61	29.90 / 45.09	45.30 / 44.74	45.87 / 38.36	43.14 / 37.74	44.50 / 52.07
GLT-T [42]	AAAI'23		48.52 / 54.29	31.74 / 56.49	52.74 / 51.43	57.60 / 52.01	44.55 / 40.69	44.42 / 54.33
MLSET [43]	RAL'23		53.20 / 58.30	33.20 / 58.60	54.30 / 52.50	53.10 / 40.90	- / -	- / -
PTTR++ [44]	PAMI'24		59.96 / 66.73	32.49 / 50.50	59.85 / 61.20	54.51 / 50.28	53.98 / 51.22	51.86 / 60.63
FlowTrack [17]	IROS'24		60.29 / 71.07	37.60 / 67.64	- / -	55.39 / 62.70	- / -	- / -
P2P [21]	IJCV'25		<u>65.15 / 72.90</u>	<u>46.43 / 75.08</u>	<u>64.96 / 65.96</u>	<u>70.46 / 66.86</u>	<u>59.02 / 56.56</u>	<u>59.84 / 72.13</u>
STTracker [22]	RAL'23	Sequence	56.11 / 69.07	37.58 / 68.36	54.29 / 60.71	48.13 / 55.40	36.31 / 36.07	49.66 / 66.77
SeqTrack3D [23]	ICRA'24		62.55 / 71.46	39.94 / 68.57	60.97 / 63.04	68.37 / 61.76	54.33 / 53.52	55.92 / 68.94
TrajTrack	Ours	Trajectory	**68.02 / 75.87**	**48.32 / 78.78**	**67.19 / 68.44**	**70.70 / 68.02**	**61.16 / 57.67**	**62.25 / 75.15**
Improvement			**↑ 2.87 / ↑ 2.97**	**↑ 1.89 / ↑ 3.70**	**↑ 2.23 / ↑ 2.48**	**↑ 0.24 / ↑ 1.16**	**↑ 2.14 / ↑ 1.11**	**↑ 2.41 / ↑ 3.02**

themselves, IoU($\mathbf{b}_t^{\text{local}}$, $\mathbf{b}_t^{\text{global}}$), as a metric for the reliability of the short-term, explicit prediction. **(1)** If the IoU is high (above a threshold λ_{IoU}), it indicates that both short-term and long-term models are in agreement. In this case, we trust the more precise, local-aware proposal, $\mathbf{b}_t^{\text{local}}$, as the final output. **(2)** If the IoU is low, it signals a potential failure of the explicit model, likely due to sparsity or occlusion. In this scenario, we rely on the more stable, long-term trajecory proposal, $\mathbf{b}_t^{\text{global}}$, as a robust fallback. This mechanism allows our tracker to be fast and precise in simple scenarios while leveraging the long-term motion prior to enhance robustness and recover from failures in challenging environments.

F. Loss Function

Our **TrajTrack** framework is trained end-to-end with a composite loss function, $\mathcal{L}_{total}$, which jointly optimizes the explicit motion proposal and the implicit trajectory prediction.
Explicit Motion Loss ($\mathcal{L}_{\text{tracking}}$). To supervise the initial proposal from Stage 1, we follow [21] and adopt the Residual Log-likelihood Estimation (RLE) loss [45], which reduces the difficulty of regression by adding a hand-designed residual term distribution to the L2 loss term, making the fitted distribution closer to the true distribution, as the tracking loss:

$$\mathcal{L}_{\text{tracking}} = \mathcal{L}_{\text{RLE}}(\mathbf{b}_t^{\text{local}}, \mathbf{b}_t^{gt}) \tag{11}$$

Implicit Trajectory Prediction Loss ($\mathcal{L}_{\text{traj}}$). To train our IMM module from Stage 2, we use the standard negative Evidence Lower Bound (ELBO) loss [46]. This loss comprises a reconstruction term, which ensures the predicted trajectory matches the ground truth, and a KL-divergence term, which regularizes the latent space:

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_{elbo} = -\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{Y},\mathbf{X})}[\log p_\theta(\mathbf{Y}|\mathbf{Z},\mathbf{X})] \tag{12}$$

$$+ \text{KL}(q_\phi(\mathbf{Z}|\mathbf{Y},\mathbf{X}) \| p_\theta(\mathbf{Z}|\mathbf{X})) \tag{13}$$

Overall Objective. The final training objective is a weighted sum of these two losses:

$$\mathcal{L}_{total} = \mathcal{L}_{\text{tracking}} + \lambda \mathcal{L}_{\text{traj}} \tag{14}$$

where λ hyper-parameter that balances the contribution of the trajectory prediction task.

IV. Experiments

Dataset and Metrics. We follow the common setup [15], [47] and conduct experiments on the large-scale nuScenes [48] dataset. Notably, due to the limited data size of the KITTI dataset (only 19 training, 2 validation sequences [13], [42]) makes it challenging to adequately evaluate the methods. In contrast, the nuScenes dataset comprises 700 training and 150 validation sequences, across 40K point cloud frames, allowing for a more comprehensive evaluation. The evaluation metrics is followed the common setup [12], [13] to report *Success* and *Precision* based on one pass evaluation (OPE) [49], [50].
Implementation Details. We follow previous works [9], [13], [21], set the previous frame $t-1$ is set as the template and the current frame t as the search region. These cropped regions, defined as $[(-4.8, 4.8), (-4.8, 4.8), (-1.5, 1.5)]$ for cars and $[(-1.92, 1.92), (-1.92, 1.92), (-1.5, 1.5)]$ for humans along the (x, y, z) axes. Additionally, the data augmentation, such as random horizontal flipping and uniform rotation within [-5°,5°] is applied to target points and bounding boxes to improve the model's generalization. The parameters of IMM are set to $H = 2, T = 12, \lambda_{IoU} = 0.5$. All experiments are conducted on NVIDIA RTX 3090 GPUs. The network is trained for 20 epochs with the AdamW optimizer, initialized with a learning rate of 0.0001, with a batch size is 32. Our experiment video is available at https://www.bilibili.com/video/BV1ahYgzmEWP.

A. Quantitative Experiment

Comparison on nuScenes: To rigorously evaluate our framework under challenging real-world conditions, we conduct comprehensive comparisons on the challenging nuScenes dataset, characterized by complex scenes and sparser point clouds from 32-beam LiDARs data in diverse scenes. Following the standard protocol for SOT evaluation, we compare exclusively against state-of-the-art SOT methods. As shown in Tab. I, our TrajTrack consistently outperforms all prior trackers across all categories, demonstrating its exceptional scalability and robustness in sparse and diverse environments. For instance, TrajTrack it surpasses a strong baseline, P2P [21], by 2.87/ 2.97% (Success/Precision) in the Car category and by an even larger margin of 1.89/3.70% in the Pedestrian category. This robust performance in a sparse environment highlights

the core advantage of our trajectory-based paradigm: by leveraging lightweight, long-term motion continuity, our method maintains a stable track where methods relying solely on instantaneous cues falter.

TABLE II: Comparison of the running speeds on different representative methods.

Method	STTracker [22]	LTTR [27]	GLT-T [42]	SeqTrack3D [23]
FPS	22.0	23.0	30.0	38.0
Method	P2B [9]	PTT [13]	PTTR++ [44]	M^2Track [15]
FPS	40.0	40.0	43.0	51.2
Method	TrajTrack (Ours)	BAT [11]	M^2Track++ [16]	**P2P** [21]
FPS	54.7	57.0	57.0	**63.9**

Running Speed: Inference speed is also a vital factor for practical applications. We present a comprehensive speed comparison of TrajTrack with other methods in Tab. II. Following common evaluation protocols [9], [12], [21], speed is measured by calculating the average running time of all frames in the Car category. On a single NVIDIA RTX 3090 GPU, TrajTrack achieves 54.7 FPS. Despite the computational overhead incurred by our IMM module, TrajTrack yields significant performance improvements, maintaining a better trade-off between accuracy and speed.

TABLE III: Ablation Study on Different Baselines

Method	**M^2-Track** [15]	**BAT** [11]
Baseline	55.85 / 65.09	40.73 / 43.29
w/ TrajTrack	58.16 / 68.23	47.07 / 51.66
improvement	↑ **2.31 / 3.14**	↑ **6.34 / 8.37**

Generalizability Across Different Trackers: To demonstrate the versatility of our trajectory-based paradigm, we replaced two distinct baselines into our framework with Implicit Motion Modeling (IMM) module. As illustrated in the Tab. III, when applied our method to the similarity-based paradigm BAT [11] and the motion-based paradigm M^2-Track [15], consistently and significantly achieving improvements of 2.31/3.14% and 6.34/8.37% in Success/Precision, respectively.This demonstrates that our approach of leveraging long-term motion continuity is a general principle that can enhance various 3D SOT architectures, regardless of their underlying paradigm.

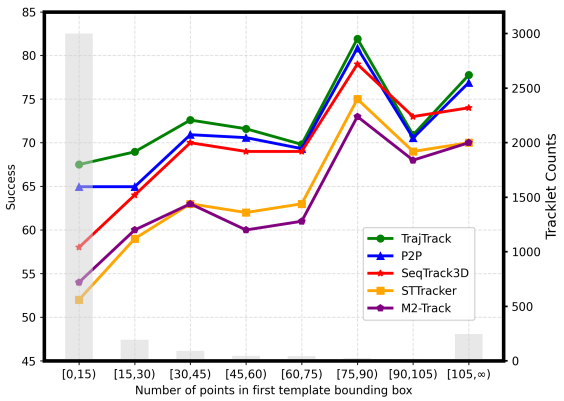

Fig. 4: **Tracking performance across varying numbers of template points in the first frame.**

Robustness to Sparsity: We evaluated our method's performance under varying levels of point cloud sparsity by analyzing its accuracy based on the number of points in the initial template. As shown in Fig. 4, TrajTrack consistently outperforms other leading methods, especially in extremely sparse scenarios. On the nuScenes dataset, where a majority of tracklets begin with fewer than 15 points, our method's advantage is most pronounced. This highlights its superior ability to leverage long-term motion continuity when instantaneous appearance cues are minimal. To further validate this, we identified a challenging subset where over half the frames contain fewer than 20 points. On these 3,061 tracklets, our method achieves a Success/Precision of 67.08% / 75.39%, significantly outperforming the baseline's 65.16% / 72.57%, which confirms its enhanced robustness in low-density conditions.

TABLE IV: Ablation Study of components in TrajTrack.

IMM Setting	**Car**		**Ped**	
	Succ.	Prec.	Succ.	Prec.
Baseline [21]	65.15	72.90	46.43	75.08
MLP-based IMM	67.27	74.14	47.68	77.33
TrajFormer-based IMM	**68.02**	**75.87**	**48.32**	**78.78**

B. Ablation Study

Ablation of IMM: We analyze the core Implicit Motion Modeling module of our method in Tab. IV. Compared to the baseline tracker [21], which relies solely on explicit two-frame motion, incorporating a simple MLP-based IMM already yields substantial performance gains, especially for the Pedestrian category (+2.25% Success). This result validates the core premise of our work: leveraging implicit motion continuity from past trajectories is highly effective. By further employing our more powerful TrajFormer-based IMM, the performance is enhanced to its peak. This systematically demonstrates that while the IMM concept itself is beneficial, the TrajFormer architecture is key to fully capturing the complex temporal dependencies in the trajectory data.

TABLE V: Ablation Study on Different timesteps Values

timesteps	Precision (%)	Success (%)
$H=1$(Baseline)	65.15	72.90
$H=2, T=12$	68.02	**75.87**
$H=3, T=6$	67.17	75.29
$H=3, T=12$	**68.07**	75.81
$H=4, T=12$	67.63	75.67

TABLE VI: Ablation Study on Different λ_{IoU} Values

λ	Precision (%)	Success (%)	Latency (ms)
0.20	66.15	74.26	17.77
0.30	66.81	74.83	17.98
0.40	67.46	75.32	18.06
0.50	68.02	75.87	18.27

Hyperparameters: We conducted a hyperparameter ablation on the historical length H, prediction horizon T, and refinement threshold λ_{IoU} (Tab. V-VI). The results show that $H=2$, $T=12$ provide the best overall performance, achieving the highest Success score while maintaining competitive Precision. This configuration offers sufficient temporal context without introducing redundant history. In addition, $\lambda_{IoU}=0.5$ yields the highest accuracy with only marginal computational overhead, and is therefore adopted as the default setting.

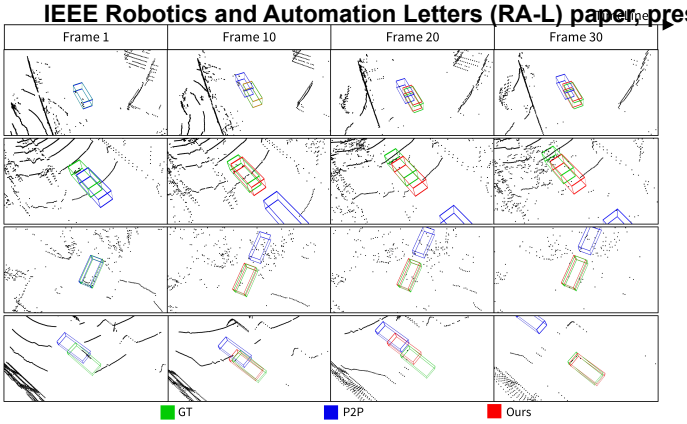

Fig. 5: **Visualization results on nuScenes.** Starting from the second frame, the global-aware trajectory proposal from Implicit Motion Modeling corrects the biased tracking results, achieving accurate tracking.

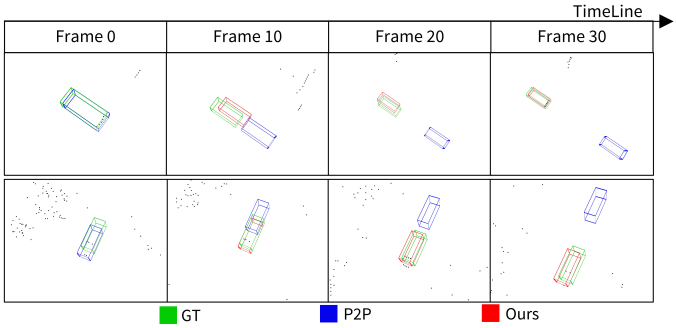

Fig. 6: **Visualization results on sparse point cloud scenes.**

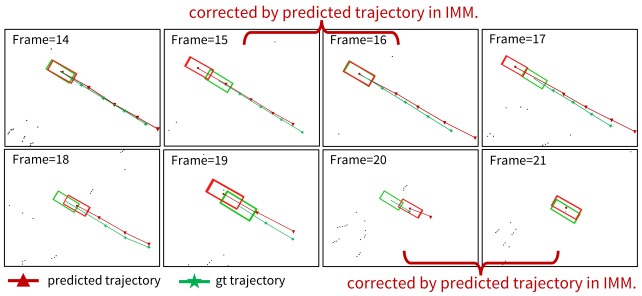

Fig. 7: **Visualization results with Implicit Trajectory Prediction correction.**

C. Visualization results.

To provide a qualitative evaluation, we visualize our tracking results against a strong baseline [21] in various challenging scenes. As shown in Fig. 5, our TrajTrack consistently demonstrates more accurate and stable tracking than the baseline. When the baseline tracker begins to fail due to occlusion or sparsity, our Trajectory-guided Proposal Refinement mechanism integrates a global-aware trajectory proposal from the Implicit Motion Modeling. This allows our method to leverage the long-term motion prior to correct misaligned bounding boxes and effectively recover the track, preventing the failure from error accumulation.Additionally, as illustrated in Fig. 7, while tracking failed, the subsequent prediction can still be corrected by leveraging long-term motion information. Furthermore, Fig. 6 highlights our method's robustness in extremely sparse scenes. Even when the target is represented by only a few scattered points, TrajTrack successfully leverages temporal consistency to track accurately. These visualizations confirm that the synergistic integration of short-term motion and long-term trajectory priors significantly enhances tracking reliability in dynamic and complex scenes.

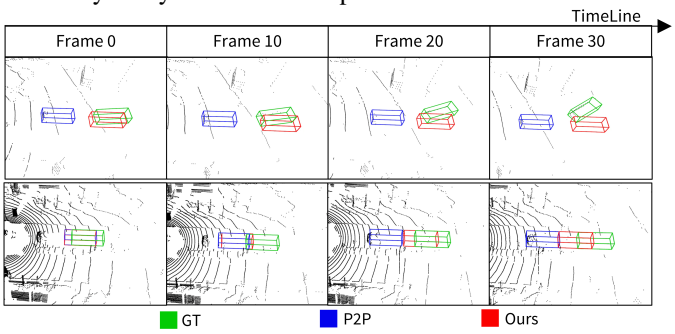

Fig. 8: **Visualization results on failed tracklets.**

V. CONCLUSIONS

In this work, we introduced **TrajTrack**, a novel trajectory-based paradigm for 3D single object tracking that resolves the critical trade-off between the robustness of two-frame methods and the efficiency of sequence-based approaches. Our key innovation is an Implicit Motion Modeling (IMM) module that leverages lightweight, historical bounding box trajectories to learn long-term motion continuity. By synergistically refining a short-term, explicit motion proposal with this powerful long-term prior, TrajTrack achieves a new state-of-the-art tracking performance in the challenging nuScenes dataset, while maintaining real-time speed. For future work, we plan to explore tighter fusion strategies between the explicit and implicit modules and investigate the application of this lightweight temporal modeling approach to other robotics perception tasks. **Limitations:** Despite outperforming baselines in highly nonlinear scenarios (e.g., sharp turns, Fig. 8), TrajTrack's reliance on motion continuity limits full convergence to the ground truth in these extreme cases. This indicates a need for more adaptive motion modeling to handle abrupt kinematic transitions. Furthermore, replacing the current heuristic proposal refinement with a learnable mechanism could yield further improvements.

REFERENCES

[1] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020.

[2] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European Conference on Computer Vision*, pp. 680–696, Springer, 2022.

[3] T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Mutr3d: A multi-camera tracking framework via 3d-to-2d queries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4537–4546, 2022.

[4] S. Ding, L. Schneider, M. Cordts, and J. Gall, "Ada-track: End-to-end multi-camera 3d multi-object tracking with alternating detection and association," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15184–15194, 2024.

[5] S. Zhou, Z. Tian, X. Chu, X. Zhang, B. Zhang, X. Lu, C. Feng, Z. Jie, P. Y. Chiang, and L. Ma, "Fastpillars: A deployment-friendly pillar-based 3d detector," *arXiv preprint arXiv:2302.02367*, 2023.

[6] S. Zhou, Z. Yuan, D. Yang, X. Hu, J. Qian, and Z. Zhao, "Pillarhist: A quantization-aware pillar feature encoder based on height-aware histogram," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27336–27345, 2025.

[7] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3d siamese tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1359–1368, 2019.

[8] Y. Cui, Z. Fang, and S. Zhou, "Point siamese network for person tracking using 3d point clouds," *Sensors*, vol. 20, no. 1, p. 143, 2019.

[9] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2b: Point-to-box network for 3d object tracking in point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6329–6338, 2020.

[10] 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 Sensors Journal*, vol. 21, no. 4, pp. 4995–5011, 2020.

[11] 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 *International Conference on Computer Vision*, pp. 13199–13208, 2021.

[12] J. Shan, S. Zhou, Z. Fang, and Y. Cui, "Ptt: Point-track-transformer module for 3d single object tracking in point clouds," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1310–1316, IEEE, 2021.

[13] S. Jiayao, S. Zhou, Y. Cui, and Z. Fang, "Real-time 3d single object tracking with transformer," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[14] 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 *Computer Vision and Pattern Recognition*, pp. 8531–8540, 2022.

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8111–8120, 2022.

[16] C. Zheng, X. Yan, H. Zhang, B. Wang, S. Cheng, S. Cui, and Z. Li, "An effective motion-centric paradigm for 3d single object tracking in point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[17] S. Li, Y. Cui, Z. Li, and Z. Fang, "Flowtrack: Point-level flow network for 3d single object tracking," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[18] S. Zhou, N. Jiahao, Z. Ziyu, C. Yichao, and L. Xiaobo, "Focustrack: One-stage focus-and-suppress framework for 3d point cloud object tracking," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

[19] S. Zhou, Y. Cao, J. Nie, Y. Fu, Z. Zhao, X. Lu, and S. Wang, "Comptrack: Information bottleneck-guided low-rank dynamic token compression for point cloud tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[20] W. Xu, S. Zhou, and Z. Yuan, "Pillartrack: Redesigning pillar-based transformer network for single object tracking on point clouds," *arXiv preprint arXiv:2404.07495*, 2024.

[21] J. Nie, F. Xie, S. Zhou, X. Zhou, D.-K. Chae, and Z. He, "P2p: Part-to-part motion cues guide a strong tracking framework for lidar point clouds," *arXiv preprint arXiv:2407.05238v2*, vol. abs/2407.05238, 2024.

[22] Y. Cui, Z. Li, and Z. Fang, "Sttracker: Spatio-temporal tracker for 3d single object tracking," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4967–4974, 2023.

[23] Y. Lin, Z. Li, Y. Cui, and Z. Fang, "Seqtrack3d: Exploring sequence information for robust 3d point cloud tracking," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6959–6965, 2024.

[24] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *ICCV*, pp. 9277–9286, 2019.

[25] L. Hui, L. Wang, M. Cheng, J. Xie, and J. Yang, "3d siamese voxel-to-bev tracker for sparse point clouds," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28714–28727, 2021.

[26] L. Hui, L. Wang, L. Tang, K. Lan, J. Xie, and J. Yang, "3d siamese transformer network for single object tracking on point clouds," in *ECCV*, 2022.

[27] Y. Cui, Z. Fang, J. Shan, Z. Gu, and S. Zhou, "3d object tracking with transformer," in *The British Machine Vision Conference*, 2021.

[28] Y. Xia, Q. Wu, W. Li, A. B. Chan, and U. Stilla, "A lightweight and detector-free 3d single object tracker on point clouds," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[29] K. Lan, H. Jiang, and J. Xie, "Temporal-aware siamese tracker: Integrate temporal context for 3d object tracking," in *Proceedings of the Asian Conference on Computer Vision*, pp. 399–414, 2022.

[30] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11522–11530, 2020.

[31] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 532–539, 2021.

[32] H. Cheng, M. Liu, L. Chen, H. Broszio, M. Sester, and M. Y. Yang, "Gatraj: A graph-and attention-based multi-agent trajectory prediction model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 205, pp. 163–175, 2023.

[33] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning. PMLR, 2021*, pp. 895–904, 2021.

[34] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15303–15312, 2021.

[35] W. Wu, X. Feng, Z. Gao, and Y. Kan, "Smart: Scalable multi-agent real-time simulation via next-token prediction," *arXiv preprint arXiv:2405.15677*, 2024.

[36] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2375–2384, 2019.

[37] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, no. CONF, 2018.

[38] W. Chen, F. Wang, and H. Sun, "S2tnet: Spatio-temporal transformer networks for trajectory prediction in autonomous driving," in *Proceedings of The 13th Asian Conference on Machine Learning*, vol. 157, pp. 454–469, 17–19 Nov 2021.

[39] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[40] H. Yadav, M. Schaefer, K. Zhao, and T. Meisen, "Caspformer: Trajectory prediction from bev images with deformable attention," in *Pattern Recognition* (A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, eds.), (Cham), pp. 420–434, Springer Nature Switzerland, 2025.

[41] Y. Lu, J. Nie, Z. He, H. Gu, and X. Lv, "Voxeltrack: Exploring multi-level voxel representation for 3d point cloud object tracking," in *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6345–6354, 2024.

[42] J. Nie, Z. He, Y. Yang, M. Gao, and J. Zhang, "Glt-t: Global-local transformer voting for 3d single object tracking in point clouds," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1957–1965, 2023.

[43] Q. Wu, C. Sun, and J. Wang, "Multi-level structure-enhanced network for 3d single object tracking in sparse point clouds," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 9–16, 2023.

[44] Z. Luo, C. Zhou, L. Pan, G. Zhang, T. Liu, Y. Luo, H. Zhao, Z. Liu, and S. Lu, "Exploring point-bev fusion for 3d point cloud object tracking with transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2024.

[45] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, and C. Lu, "Human pose regression with residual log-likelihood estimation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11005–11014, 2021.

[46] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[47] Z. Hu, S. Zhou, S. Zhao, and Z. Yuan, "Mvctrack: Boosting 3d point cloud tracking via multimodal-guided virtual cues," 2024.

[48] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[49] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, 2013.

[50] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.