State Evolution Awareness for Category-agnostic 3D Point Cloud Tracking

Xiantao Hu[1] **Ying Tai**[2*] **Jinxia Xie**[1] **Bineng Zhong**[3]
Guangwei Gao[1] **Jianjun Qian**[1] **Jian Yang**[1*]
[1]Nanjing University of Science and Technology [2]Nanjing University
[3]Key Laboratory of Education Blockchain and Intelligent Technology
(Guangxi Normal University), Ministry of Education
{xiantaohu,jinxia_xie,gwgao,csjqian,csjyang}@njust.edu.cn
yingtai@nju.edu.cn bnzhong@gxnu.edu.cn

Abstract

Existing 3D point cloud tracking methods have not fully exploited temporal information in point cloud sequences, making it difficult to effectively capture the continuous evolution of target states and thereby limiting tracking stability in scenarios with large cross-category geometric variation. To address this issue, we propose TETrack3D, a framework for category-agnostic 3D point cloud tracking. The core idea of TETrack3D is to elevate temporal information from an auxiliary cue to an explicit learning constraint for cross-frame association. Specifically, we introduce a historical feature reuse mechanism to preserve and reuse target-related features accumulated over multiple frames, enabling the association process in the current frame to access richer and finer-grained historical context. We further design a state evolution supervision module, which guides the model to learn temporally consistent target motion patterns from historical observations by predicting future target states. In addition, to alleviate cross-frame feature drift caused by target-state changes in point cloud sequences, we propose a temporal distribution alignment strategy based on optimal transport theory, which constrains target features across adjacent frames at the distribution level and improves the stability of temporal target modeling. Extensive comparative experiments on KITTI, nuScenes, and Waymo demonstrate that explicitly modeling the target-state evolution process effectively improves the localization robustness of 3D object tracking. Code is available at https://github.com/NJU-PCALab/TETrack3D.

1 Introduction

3D point cloud single object tracking (3D SOT) [2, 12, 16] is important for autonomous driving [3, 24], robotics [33], and embodied perception [40]. While recent methods achieve strong results, many remain category-specific [10, 14, 36], which limits scalability in open-world settings. Category-agnostic tracking [17, 26] is therefore more practical, but also more challenging because the tracker must generalize across large geometric variation.

As shown in Fig. 1, existing methods mainly follow two paradigms. Frame-by-frame matching [18, 20, 32] localizes targets through local template-search association, but is fragile under sparsity, occlusion, and unstable observations. Temporal-enhanced methods [9, 26, 28, 31] improve robustness with historical cues, yet they usually treat temporal information as auxiliary enhancement rather than an explicit constraint on localization. As a result, historical observations are not reliably converted into stable state priors when target motion or geometry changes significantly.

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Although the above methods focus on frame-to-frame matching or the use of historical information, neither explicitly models the temporal evolution of target states in point cloud sequences. For category-agnostic 3D tracking, directly learning a unified representation that generalizes across different target categories remains highly challenging. In contrast, characterizing the temporal evolution trend of target states from continuous observations is typically more transferable, since it relies less on category-specific semantics. In this work, we regard such temporal variation patterns as a *target-state evolution process*. We argue that the lack of explicit modeling of this process is a key bottleneck that limits existing methods from achieving high-performance category-agnostic 3D tracking. Based on this, the limitation can be further summarized into the following three aspects. First, *historical target evidence is insufficiently preserved*. Existing methods often aggregate, compress, or contextualize historical observations before injecting them into the current-frame representation, which may lose fine-grained target evidence and thus provide limited support when observations change drastically. Second, *temporal consistency lacks explicit supervision*. Most temporal modeling strategies mainly use historical information to enhance current representations, but do not explicitly constrain the continuous evolution of the target state over time, making it difficult to learn stable temporal consistency. Third, *stable temporal feature evolution is difficult to maintain*. Due to sparse sampling, occlusion, and viewpoint changes, target point clouds in adjacent frames often exhibit substantial observation shifts and feature distribution discrepancies. Such feature drift disrupts the temporal consistency of target-related representations and weakens the stability of target evolution modeling over time.

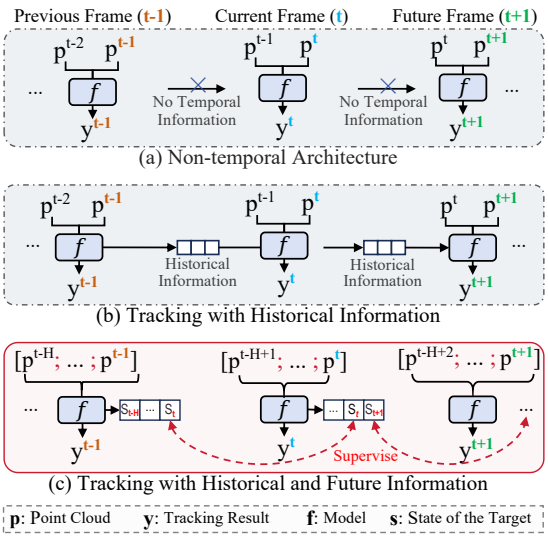

Figure 1: Three paradigms for 3D point cloud tracking. (a) Frame-to-frame matching associates targets mainly through local matching. (b) Temporal-enhanced tracking introduces historical context as auxiliary information. (c) TETrack3D explicitly models target evolution to enable more robust association.

To realize this idea, TETrack3D introduces three tightly coupled designs for modeling target-state evolution, addressing complementary aspects of historical evidence preservation, temporal dependency learning, and cross-frame stability. Specifically, *historical feature reuse* preserves and propagates target-related evidence accumulated across frames, enabling the current frame to access richer and finer-grained historical cues during association. Building upon this, *state evolution supervision* introduces future-state prediction as an explicit training signal, encouraging the model to learn temporally coherent target dynamics from historical observations rather than relying solely on instantaneous matching cues. To further enhance robustness, *temporal distribution alignment* mitigates cross-frame feature drift caused by sparse and unstable point cloud observations by aligning target-related feature distributions across adjacent frames via optimal transport, thereby stabilizing the evolution of feature distributions over time. Notably, both state evolution supervision and temporal distribution alignment are only applied during training, introducing negligible overhead during inference.

- We propose TETrack3D, a unified framework for category-agnostic 3D point cloud tracking that revisits cross-frame association from the perspective of target evolution modeling.
- We develop an evolution-aware temporal learning framework for explicit target-state evolution modeling, enabled by historical feature reuse, state evolution supervision, and temporal distribution alignment.
- Extensive experiments on KITTI, nuScenes, and Waymo demonstrate that TETrack3D achieves strong performance with negligible additional inference overhead.

2 Related work

2.1 3D point cloud tracking

Existing 3D SOT methods mainly perform target localization through frame-level association, including template-search matching [4, 20] and short-term motion estimation [18, 36, 39]. These approaches rely on appearance similarity or relative motion between consecutive frames, and are typically trained in a category-specific manner, requiring separate models for different object categories. Recent efforts have moved toward category-unified 3D tracking. SiamCUT and MoCUT [17] first explore unified tracking across categories by mitigating cross-category geometric variations through adaptive representation learning, unified inputs, and learning objectives. TrackAny3D [26] further advances this direction by transferring pretrained 3D priors and introducing temporal context optimization for improved generalization. Beyond category unification, several works incorporate historical or sequential information to improve tracking robustness. For example, MBPTrack [31], SeqTrack3D [9], and HVTrack [28] propagate historical cues or sequence-level information to alleviate sparse observations, occlusion, and temporal variation. Different from these works, TETrack3D focuses on explicitly modeling the target-state evolution process from historical observations and leveraging it to guide cross-frame association in category-agnostic 3D tracking.

2.2 Temporal modeling in tracking

Temporal modeling has become a key component in 2D, 3D, and multi-modal tracking, aiming to improve cross-frame association robustness under appearance variations, occlusions, and motion uncertainty. Early approaches mainly rely on memory retrieval [19, 28] or template updating [23] to preserve historical appearance information. More recent works introduce mechanisms such as token propagation [5, 26, 29] and autoregressive queries [9] to model temporal dependencies across frames. In addition, some methods explicitly process multiple frames to capture richer temporal information, such as ODTrack [37] and TATtrack [25]. Different from these approaches, our method focuses on modeling the target-state evolution process from historical observations.

3 Method

3.1 Overview

Task Definition. Following recent category-agnostic 3D tracking methods [18, 26], we adopt the standard template-search tracking setting and focus on improving target localization with temporal target modeling. At time step t, the tracker uses a template region P_{t-1}^{tmp} from the previous frame and a search region P_t^{src} from the current frame as the basic inputs. Beyond this standard setting, TETrack3D additionally preserves and reuses target-related evidence accumulated over historical frames, enabling the model to capture target-state evolution from continuous observations. Based on these inputs, the tracker predicts a 4-DoF offset $\Delta b_t = (\Delta x_t, \Delta y_t, \Delta z_t, \Delta\theta_t)$ relative to the previous target box, while the target size is inherited from the initialized box. The final prediction is obtained by transforming the previous box with the estimated offset.

Framework Overview. As shown in Fig. 2, TETrack3D follows the template-search tracking pipeline. Specifically, we encode the template and search point clouds using RECON [21], and decode the resulting search features with an RPN-based localization head. On top of this baseline, we introduce three components for temporal target modeling: (1) **historical feature reuse**, which preserves and reuses target-related evidence accumulated over historical frames; (2) **state evolution supervision**, a training-only component that uses future-state prediction to learn temporally consistent target motion patterns from historical observations; and (3) **temporal distribution alignment**, a training-only strategy that aligns target-related feature distributions across adjacent frames via optimal transport to improve temporal consistency.

3.2 Historical feature reuse

Different from conventional frame-wise tracking methods [17, 26], our framework preserves target-related evidence from historical template frames within the backbone, allowing the current frame to access richer historical context during target localization.

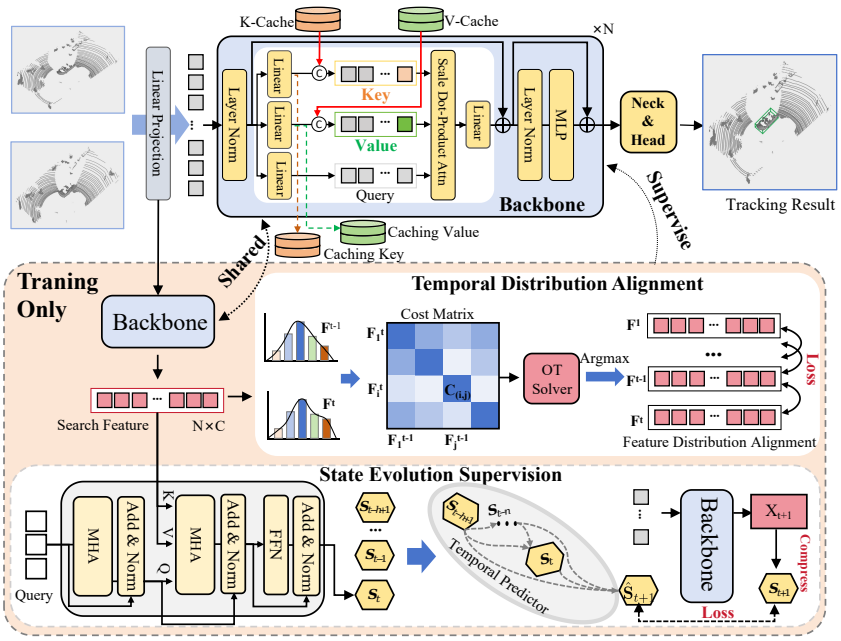

Figure 2: **Overall architecture of TETrack3D**. The proposed TETrack3D explicitly models target-state evolution through historical feature reuse, state evolution supervision, and temporal distribution alignment, where the latter two are only used during training.

Specifically, for each transformer block l in backbone, we maintain a historical memory $\mathcal{C}_t^l = \{\mathcal{K}_t^l, \mathcal{V}_t^l\}$, which stores the historical key and value sets available at step t. During attention computation, the current template-search pair attends not only to the current tokens, but also to the historical template tokens stored in the memory. Let $\mathbf{Q}_t^l$, $\mathbf{K}_t^l$, and $\mathbf{V}_t^l$ denote the queries, keys, and values of the current template-search pair at block l. The attention output is computed as

$$\mathbf{Y}_t^l = \mathrm{Attn}\big(\mathbf{Q}_t^l,\ [\mathcal{K}_t^l; \mathbf{K}_t^l],\ [\mathcal{V}_t^l; \mathbf{V}_t^l]\big)\,, \tag{1}$$

where $\mathcal{K}_t^l$ and $\mathcal{V}_t^l$ denote the historical key and value sets retrieved from the memory at step t, and $[\cdot;\cdot]$ represents concatenation along the token dimension. In this way, the current frame can directly reuse fine-grained target-related evidence accumulated from multiple previous template observations, rather than relying only on the current template.

After the current step is processed, the template tokens are written into the memory through

$$\mathcal{C}_{t+1}^l = \mathrm{Update}\big(\mathcal{C}_t^l, \bar{\mathbf{K}}_t^l, \bar{\mathbf{V}}_t^l\big)\,, \tag{2}$$

where $\bar{\mathbf{K}}_t^l$ and $\bar{\mathbf{V}}_t^l$ denote the keys and values of the template tokens at step t, and $\mathrm{Update}(\cdot)$ denotes the memory update operation. In our design, only template tokens are written into the memory, which provides a more stable target-oriented reference for historical feature reuse.

We keep the earliest template frame as a fixed reference and update the remaining memory slots in a first-in-first-out manner with newly arrived template tokens. This design preserves long-term target cues while incorporating recent observations, making the memory adaptive to target-state evolution.

3.3 State evolution supervision

While historical feature reuse enriches the current representation with target-related evidence accumulated from previous template observations, it mainly serves as a history-aware feature enhancement mechanism. To explicitly model target-state evolution over time, we further introduce a training-only state compression and temporal prediction branch.

Given the backbone network output features of the current search region, denoted by $\mathbf{Z}_t$, we use a small set of learnable state queries to extract compact state tokens from dense search tokens.

Specifically, the learnable queries attend to $\mathbf{Z}_t$ through cross-attention to gather target-relevant information. The resulting outputs are compact state tokens $\mathbf{S}_t$, which summarize the target state at frame t. In this way, dense search tokens are distilled into compact state tokens for subsequent state evolution supervision.

Based on the compact state tokens, we maintain a history queue containing the most recent observed states:

$$\Psi_t = [\mathbf{S}_{t-h+1}, \ldots, \mathbf{S}_t], \tag{3}$$

where h denotes the maximum history length. At the beginning of a sequence, only the available recent states are retained in the queue. We then employ a temporal predictor $G(\cdot)$ to model dependencies in the observed state sequence. As shown in Fig. 2, $G(\cdot)$ is implemented with a mamba architecture and takes the history queue as input to predict the compact state tokens of the next frame:

$$\hat{\mathbf{S}}_{t+1} = G(\Psi_t), \tag{4}$$

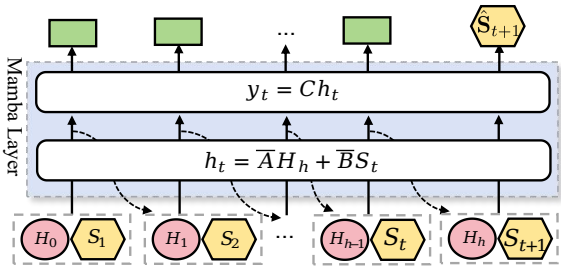

Figure 3: **Temporal predictor in state evolution supervision**. Implemented with a mamba-based sequence model, the predictor takes historical states as input and predicts the future state $\hat{\mathbf{S}}_{t+1}$.

where $\hat{\mathbf{S}}_{t+1}$ has the same token number and feature dimension as $\mathbf{S}_{t+1}$. Since the predictor has no access to the search features of frame $t+1$, it is encouraged to capture the temporal regularities of target-state evolution from the observed state sequence, rather than relying on instantaneous cues from the next frame.

After frame $t+1$ is encoded, we apply the same state-query-based compressor to its backbone output features to obtain the observed compact state tokens $\mathbf{S}_{t+1}$, which serve as the supervision target for the prediction $\hat{\mathbf{S}}_{t+1}$. The state evolution loss is defined as

$$\mathcal{L}_{\text{state}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \text{SmoothL1}\left(\hat{\mathbf{S}}_{t+1}, \mathbf{S}_{t+1}\right), \tag{5}$$

where T denotes the frame sequence length. By enforcing the predicted future state tokens to match the observed ones, the proposed supervision encourages the backbone and the state queries to encode temporally coherent target dynamics from continuous observations. Although this branch is only used during training, it improves the quality of the learned features and thus benefits target localization without introducing additional inference overhead.

3.4 Temporal distribution alignment

Although state evolution supervision explicitly constrains target dynamics through future-state prediction, robust temporal modeling also requires temporally consistent target-related token features in the search region. In practical point cloud sequences, however, sparsity, occlusion, and viewpoint variation often cause the features of target-related search tokens to vary over time, making it difficult for the model to learn stable temporal distributions. To alleviate this issue, we introduce a training-only temporal distribution alignment module based on optimal transport, which regularizes the distributions of target-related search tokens over time.

For two consecutive time steps, let

$$\mathbf{F}^t = \{f_i^t\}_{i=1}^N, \qquad \mathbf{F}^{t+1} = \{f_j^{t+1}\}_{j=1}^N \tag{6}$$

denote the sampled search-token feature sets at steps t and $t+1$, respectively, where N is the number of sampled tokens at each time step. Based on these features, we define the pairwise transport cost as

$$C_{i,j} = d(f_i^t, f_j^{t+1}), \tag{7}$$

where $C_{i,j}$ measures the matching cost between token f_i^t at step t and token f_j^{t+1} at step $t+1$, and $d(\cdot,\cdot)$ denotes the feature distance.

To emphasize target-related regions and suppress background interference, each token is assigned a nonnegative foreground weight. These weights are then normalized into two foreground-aware token distributions:

$$p_i^t = \frac{w_i^t}{\sum_{i'} w_{i'}^t}, \qquad p_j^{t+1} = \frac{w_j^{t+1}}{\sum_{j'} w_{j'}^{t+1}}, \tag{8}$$

where w_i^t and w_j^{t+1} denote the foreground weights of the corresponding tokens at steps t and $t+1$, respectively. Here, p^t and p^{t+1} represent the normalized foreground-aware distributions over the token sets of the two time steps, so that target-related tokens contribute more to the transport process.

Given the cost matrix C and the foreground-aware token distributions p^t and p^{t+1}, we compute the transport plan using the Sinkhorn-based optimal transport (OT) solver:

$$\Gamma = \mathrm{Sinkhorn}(C, p^t, p^{t+1}), \tag{9}$$

where Γ is the soft transport matrix, and each element $\Gamma_{i,j}$ represents the transport mass assigned between token f_i^t and token f_j^{t+1}. In this way, the transport plan establishes a soft distribution-level correspondence between target-related token sets at adjacent time steps.

Based on the resulting transport plan, the temporal distribution alignment loss is defined as

$$\mathcal{L}_{\mathrm{tda}} = \frac{1}{|\mathcal{P}|} \sum_{t \in \mathcal{P}} \langle \Gamma^t, C^t \rangle, \tag{10}$$

where $\mathcal{P}$ denotes the set of valid adjacent time-step pairs with non-empty foreground mass in both steps, $|\mathcal{P}|$ is the number of such pairs, and $\langle \Gamma^t, C^t \rangle$ is the transport cost for the pair $(t, t+1)$. By minimizing this loss, the model is encouraged to learn temporally consistent target-related feature distributions, thereby reducing feature drift over time and providing a more stable feature-level constraint for target-state evolution modeling. Since this module is only used during training, it introduces no additional inference overhead.

Table 1: Comparison with SOTA on KITTI [3] benchmark. **Bold** and underline denote current and previous best performance.

Type	Method	Car [6,424]	Pedestrian [6,088]	Van [1,248]	Cyclist [308]	Mean [14,068]
Category-specific	P2B [20]	56.2 / 72.8	28.7 / 49.6	40.8 / 48.4	32.1 / 44.7	42.4 / 60.0
	V2B [6]	70.5 / 81.3	48.3 / 73.5	50.1 / 58.0	40.8 / 49.7	58.4 / 75.2
	PTTR [38]	65.2 / 77.4	50.9 / 81.6	52.5 / 61.8	65.1 / 90.5	57.9 / 78.2
	STNet [7]	72.1 / 84.0	49.9 / 77.2	58.0 / 70.6	73.5 / 93.7	61.3 / 80.1
	GLT-T [16]	68.2 / 82.1	52.4 / 78.8	52.6 / 62.9	68.9 / 92.1	60.1 / 79.3
	CXTrack [30]	69.1 / 81.6	67.0 / 91.5	60.0 / 71.8	74.2 / 94.3	67.5 / 85.3
	M^2Track [36]	65.5 / 80.8	61.5 / 88.2	53.8 / 70.7	73.2 / 93.5	62.9 / 83.4
	SyncTrack [15]	73.3 / 85.0	54.7 / 80.5	60.3 / 70.0	73.1 / 93.8	64.1 / 81.9
	MBPTrack [31]	73.4 / 84.8	68.6 / 93.9	**61.3** / 72.7	76.7 / 94.3	70.3 / 87.9
	PTTR++ [14]	73.4 / 84.5	55.2 / 84.7	55.1 / 62.2	71.6 / 92.8	63.9 / 82.8
	M3SOT [10]	**75.9** / **87.4**	66.6 / 92.5	59.4 / 74.7	70.3 / 93.4	70.3 / **88.6**
	StreamTrack [13]	72.6 / 83.7	**70.5** / **94.7**	61.0 / **76.9**	**78.1** / **94.6**	**70.8** / 88.1
Category-agnostic	STNet [7]	69.5 / 80.5	43.8 / 67.4	58.1 / 68.2	73.8 / 94.0	57.5 / 74.0
	CXTrack [30]	60.2 / 72.6	54.6 / 81.6	57.6 / 70.0	44.4 / 57.0	57.2 / 75.9
	M^2Track [36]	62.9 / 77.0	57.4 / 84.4	66.2 / 80.7	76.0 / 93.7	61.1 / 80.9
	MBPTrack [31]	62.3 / 72.1	50.2 / 80.9	66.6 / 78.2	71.8 / 92.2	56.1 / 74.9
	SiamCUT [17]	58.1 / 73.9	48.2 / 76.2	63.1 / 74.9	36.7 / 47.4	54.0 / 74.6
	MoCUT [17]	67.6 / 80.5	63.3 / 90.0	64.5 / 78.8	**76.7** / 94.2	65.8 / 85.0
	TrackAny3D [26]	73.4 / 85.2	59.6 / 85.6	70.0 / 82.8	74.7 / 94.0	67.1 / 85.4
	TETrack3D	**74.4** / **85.7**	**64.5** / **90.3**	**73.9** / **86.0**	76.6 / **94.6**	**70.2** / **87.9**

Table 2: Comparison with SOTA on nuScenes [1] benchmark. **Bold** and underline denote current and previous best performance.

Type	Method	Car [64,159]	Pedestrian [33,227]	Truck [13,587]	Trailer [3,352]	Bus [2,953]	Mean [117,278]
Category-specific	SC3D [4]	22.3 / 21.9	11.3 / 12.7	30.7 / 27.7	35.3 / 28.1	29.4 / 24.1	20.7 / 22.2
	P2B [20]	38.8 / 43.2	28.4 / 52.2	43.0 / 41.6	49.0 / 40.1	33.0 / 27.4	36.5 / 45.1
	PTT [22]	41.2 / 45.3	19.3 / 32.0	50.2 / 48.6	51.7 / 46.5	39.4 / 36.7	36.3 / 41.7
	PTTR [38]	51.9 / 58.6	29.9 / 45.1	45.3 / 44.7	45.9 / 38.4	43.1 / 37.7	44.5 / 52.1
	GLT-T [16]	48.5 / 54.3	31.7 / 56.5	52.7 / 51.4	57.6 / 52.0	44.6 / 40.7	44.4 / 54.3
	CXTrack [30]	42.6 / 48.0	32.8 / 59.3	40.3 / 38.7	51.5 / 41.4	38.8 / 30.7	39.7 / 49.5
	M^2Track [36]	55.9 / 65.1	32.1 / 60.7	57.4 / 59.5	57.6 / 58.3	51.4 / 51.4	49.3 / 62.7
	SeqTrack3D [9]	**62.6 / 71.5**	39.9 / 68.6	61.0 / 63.0	**68.4 / 61.8**	54.3 / 53.5	55.9 / 68.9
	MBPTrack [31]	62.5 / 70.4	45.3 / 74.0	62.2 / 63.3	65.1 / 61.3	55.4 / 51.8	57.5 / 69.9
	PTTR++ [14]	60.0 / 66.7	32.5 / 50.5	59.9 / 61.2	54.5 / 50.3	54.0 / 51.2	51.9 / 60.6
	SCVTrack [34]	58.9 / 67.7	34.5 / 61.5	60.6 / 61.4	59.5 / 60.1	54.3 / 53.6	52.1 / 64.7
	StreamTrack [13]	62.1 / 70.8	38.4 / 68.6	**64.7 / 66.6**	66.7 / 64.3	60.7 / 59.7	55.8 / 69.2
Category-agnostic	CXTrack [30]	43.4 / 47.4	34.2 / 61.5	48.2 / 44.8	54.5 / 42.4	43.2 / 35.4	41.7 / 50.7
	M^2Track	50.8 / 57.6	28.4 / 53.5	56.0 / 57.4	47.9 / 42.1	51.7 / 49.1	45.0 / 55.8
	MBPTrack [31]	55.8 / 61.5	37.6 / 65.4	59.0 / 57.2	58.1 / 46.3	58.4 / 54.2	51.1 / 61.5
	SiamCUT [17]	41.0 / 44.9	31.4 / 53.8	53.9 / 52.7	63.3 / 58.2	41.0 / 38.0	40.4 / 48.5
	MoCUT [17]	57.3 / 66.0	33.5 / 63.1	61.8 / 64.4	60.9 / 61.8	57.4 / 56.1	51.2 / 64.6
	TrackAny3D [26]	59.3 / 66.5	40.4 / 68.7	62.7 / 62.8	66.1 / 59.2	61.0 / 58.0	54.6 / 66.3
	TETrack3D	**61.1 / 69.4**	**42.3 / 73.1**	**66.2 / 67.3**	**67.6 / 59.7**	**64.5 / 61.8**	**56.6 / 69.7**

4 Experiments

4.1 Implementation details

We follow TrackAny3D [26] for dataset preprocessing and evaluation. For each tracklet, the template and search regions are cropped around the target, and each cropped point cloud is randomly resampled to 1024 points. During training, each sample is formed by sampling 8 frames from the same tracklet to construct a sequence of consecutive template-search pairs. Our model is built on the pretrained RECON encoder [21]. The backbone contains 12 Transformer blocks with feature dimension 384, 6 attention heads, 128 groups, and group size 32. To reuse historical target-related evidence, we maintain a historical memory of length 4 in the backbone. One slot is reserved for the fixed initial template as a stable anchor for long-term target identity, while the remaining three slots are updated in a first-in-first-out manner to preserve recent observations. For state evolution supervision, we set the state history length to 3 and use a lightweight two-layer Mamba module as the temporal predictor to forecast the target state at the next time step. The model is trained with the AdamW optimizer [11] for 60 epochs using a batch size of 32. The initial learning rate is set to 1×10^{-4} and decayed with a cosine annealing schedule. We follow the conventional training losses used in previous methods [26], and further introduce two additional training objectives, namely the state evolution loss and the temporal distribution alignment loss. The weights of these two losses are both set to 0.2. During inference, state evolution supervision and temporal distribution alignment are disabled, and only the base tracker with historical memory is retained. Training uses eight NVIDIA RTX 4090 GPUs with 48 GB memory per GPU and two Intel Xeon Silver 4314 CPUs; desktop inference is evaluated on a single RTX 4090 GPU.

4.2 Comparison with state-of-the-art methods

KITTI. Table 1 shows that TETrack3D achieves the best category-agnostic performance on KITTI [3], reaching 70.2 / 87.9 in mean Success / Precision. It surpasses TrackAny3D [26] by 3.1% / 2.5% and MoCUT [17] by 4.4% / 2.9%. Despite using a single model across categories, TETrack3D remains competitive with leading category-specific trackers. Different from TrackAny3D [26], which addresses category-agnostic tracking mainly through stronger generalized target representations, TETrack3D builds on target evolution, a temporal structure that is inherently less category-dependent and thus more transferable across object types. Through historical feature reuse and training-only temporal supervision, TETrack3D learns more temporally coherent target representations and remains robust under target state changes.

Table 3: Comparison with SOTA on WOD [24] benchmark. **Bold** and underline denote current and previous best performance.

Type	Method	Vehicle [185,731]			
		Easy	Medium	Hard	Mean
Category-specific	P2B [20]	57.1 / 65.4	52.0 / 60.7	47.9 / 58.5	52.6 / 61.7
	BAT [35]	61.0 / 68.3	53.3 / 60.9	48.9 / 57.8	54.7 / 62.7
	V2B [6]	64.5 / 71.5	55.1 / 63.2	52.0 / 62.0	57.6 / 65.9
	STNet [7]	65.9 / 72.7	57.5 / 66.0	54.6 / 64.7	59.7 / 68.0
	TAT [8]	66.0 / 72.6	56.6 / 64.2	52.9 / 62.5	58.9 / 66.7
	CXTrack [30]	63.9 / 71.1	54.2 / 62.7	52.1 / 63.7	57.1 / 66.1
	M^2Track [36]	68.1 / 75.3	**58.6 / 66.6**	55.4 / 64.9	61.1 / 69.3
	MBPTrack [31]	**68.5 / 77.1**	58.4 / 68.1	**57.6 / 69.7**	**61.9 / 71.9**
	SiamDisst [27]	-	-	-	61.2 / 70.5
	MemDisst [27]	-	-	-	**61.9 / 71.9**
Category-agnostic	CXTrack [30]	60.7 / 67.6	50.5 / 57.7	46.0 / 55.8	52.8 / 60.7
	STNet [7]	67.5 / 75.2	59.2 / 68.2	55.6 / 66.4	61.1 / 70.2
	MBPTrack [31]	65.6 / 74.4	55.4 / 64.6	52.5 / 63.2	58.2 / 67.7
	SiamCUT [17]	58.3 / 66.0	50.8 / 60.8	49.2 / 59.1	53.0 / 62.2
	MoCUT [17]	68.3 / 75.0	59.4 / 66.9	57.1 / 66.3	61.9 / 69.7
	TrackAny3D [26]	72.6 / 80.2	60.5 / 69.6	57.5 / 68.9	64.0 / 73.3
	TETrack3D	**78.4 / 82.6**	**69.9 / 76.2**	**67.2 / 76.2**	**72.2 / 78.6**

nuScenes. Compared with KITTI, nuScenes [1] involves greater variation in object categories, motion patterns, and scene layouts, which makes category-agnostic tracking more challenging. As shown in Table 2, TETrack3D achieves 56.6 / 69.7 in mean Success / Precision, outperforming the previous best category-agnostic tracker TrackAny3D [26] by 2.0% / 3.4%. It also consistently exceeds other category-agnostic methods such as MoCUT [17] and MBPTrack [31], and even surpasses most category-specific trackers across categories. These results indicate that explicit target evolution modeling provides a more effective temporal learning signal than methods that mainly enhance temporal cues at the feature level, leading to more stable tracking under larger cross-frame variation.

WOD. Different from KITTI and nuScenes, WOD [24] exhibits a larger distribution shift from the training setting, and therefore serves as a more direct test of cross-dataset generalization. As shown in Table 3, TETrack3D achieves 72.2 / 78.6 in mean Success / Precision, surpassing TrackAny3D [26] by 8.2% / 5.3%. The advantage becomes even larger on the more challenging splits, where TETrack3D reaches 69.9 / 76.2 on the medium split and 67.2 / 76.2 on the hard split, exceeding TrackAny3D by 9.4% / 6.6% and 9.7% / 7.3%, respectively. Compared with other category-agnostic trackers such as MoCUT [17] and MBPTrack [31], the gap is also more pronounced on this benchmark. These results indicate that TETrack3D acquires the ability to perceive target evolution, which captures temporal structure that is more transferable across datasets and therefore yields stronger generalization under distribution shift.

4.3 Ablation study

Model Components. Table 4 summarizes the contribution of each component. The full model achieves the best result of 70.2 / 87.9. Removing historical feature reuse causes the largest drop of 5.6% / 4.2%, to 64.6 / 83.7, since the tracker then reduces to a much weaker frame-wise matching scheme without access to fine-grained historical target evidence. Removing temporal distribution alignment leads to a drop of 3.7% / 1.8%, to 66.5 / 86.1, suggesting that distribution-level alignment provides an important complementary constraint by maintaining stable temporal feature distributions. Removing state evolution supervision causes a 1.6% / 3.4% drop, to 68.6 / 84.5, indicating that merely introducing temporal information is insufficient, and that explicit future-state supervision is needed to turn target evolution into a learnable training signal. We also observe that removing the historical memory or the fixed initial anchor consistently hurts performance, with drops of 2.0% / 2.9% and 1.7% / 1.3%, respectively, validating the proposed fixed-plus-FIFO design for balancing long-term identity cues and recent temporal evidence. The three components therefore contribute in a complementary manner to target evolution modeling.

Table 4: Component ablations of TETrack3D on KITTI benchmark.

Variant	Car	Ped.	Van	Cyc.	Mean
w/o Historical Feature Reuse	72.5 / 84.6	56.2 / 84.3	63.9 / 74.5	66.9 / 92.1	64.6 / 83.7
w/o State Evolution Supervision	73.4 / 84.1	63.2 / 85.9	67.7 / 77.5	79.0 / 95.1	68.6 / 84.5
w/o Temporal Distribution Alignment	72.2 / 85.2	58.8 / 86.8	73.4 / 85.3	71.9 / 93.3	66.5 / 86.1
w/o Historical Memory	71.4 / 83.3	63.5 / 86.6	72.6 / 83.9	75.5 / 93.8	68.2 / 85.0
w/o Fixed Anchor	73.7 / 85.1	61.9 / 88.6	71.3 / 83.0	78.2 / 94.5	68.5 / 86.6
Full Model	**74.4 / 85.7**	**64.5 / 90.3**	**73.9 / 86.0**	**76.6 / 94.6**	**70.2 / 87.9**

Number of Historical States. Since the history length determines how much temporal context the temporal predictor can leverage for future-state prediction, we study its effect in Table 5. Increasing the history length from 1 to 3 improves the mean performance from 67.2 / 85.2 to 70.2 / 87.9, with a gain of 3.0% / 2.7%, indicating that target evolution cannot be reliably inferred from only one or two past states. However, further increasing the history length to 4 causes the performance to drop to 69.3 / 87.1, suggesting that overly long state sequences may introduce redundant or noisy dynamics that weaken the prediction target. We therefore set the history length to 3 in the final model.

Table 5: Ablation of the number of historical states used for future-state prediction on KITTI.

Hist. States	Car	Ped.	Van	Cyc.	Mean
1	72.4 / 84.0	60.0 / 86.0	72.7 / 85.1	76.4 / 94.2	67.2 / 85.2
2	73.8 / 85.3	61.6 / 87.5	72.3 / 85.2	73.5 / 93.6	68.4 / 86.4
3	**74.4 / 85.7**	**64.5 / 90.3**	**73.9 / 86.0**	**76.6 / 94.6**	**70.2 / 87.9**
4	73.2 / 85.7	64.4 / 88.7	72.5 / 85.0	71.5 / 93.7	69.3 / 87.1

Number of Historical Reference Frames. The memory size controls the trade-off between long-term identity cues and recent temporal evidence in historical feature reuse. We therefore study its effect in Table 6. Using only the fixed initial anchor yields 68.2 / 85.0, indicating that a single static template cannot adequately support tracking under significant target variation. As the number of historical reference frames increases, the performance consistently improves and reaches the best result of 70.2 / 87.9 when the memory length is 4, corresponding to one fixed initial anchor and three dynamically updated historical frames. This suggests that evolution-aware association benefits from both a stable long-term anchor and a small set of recent observations that capture ongoing target changes. However, further enlarging the memory to 5 frames causes the performance to drop to 69.3 / 87.1, suggesting that excessive historical references introduce redundancy and noise rather than useful new evidence. We therefore set the memory length to 4 in the final design.

Table 6: Ablation of the number of historical reference frames retained in the backbone memory on KITTI, where length 1 keeps only the fixed initial anchor.

Hist. Ref.	Car	Ped.	Van	Cyc.	Mean
1	71.4 / 83.3	63.5 / 86.6	72.6 / 83.9	75.5 / 93.8	68.2 / 85.0
2	74.7 / 86.0	60.7 / 84.7	74.0 / 85.9	78.4 / 94.3	68.7 / 85.6
3	72.8 / 84.1	65.5 / 89.3	73.0 / 84.7	78.4 / 95.0	69.8 / 86.7
4	**74.4 / 85.7**	**64.5 / 90.3**	**73.9 / 86.0**	**76.6 / 94.6**	**70.2 / 87.9**
5	73.2 / 85.7	64.4 / 88.7	72.5 / 85.0	71.5 / 93.7	69.3 / 87.1

Qualitative Comparison. Figure 4 provides qualitative comparisons of predicted bounding boxes in challenging cases. Since TrackAny3D [26] lacks an explicit perception of target motion, it is easily misled by similar surrounding distractors and drifts to wrong targets. In contrast, TETrack3D explicitly models target evolution through historical feature reuse and training-only temporal supervision, which equips the tracker with target evolution awareness and helps it remain locked on the correct target even under strong distractor interference, sparse observations, or partial occlusion. These examples show that the proposed evolution-aware design improves not only the average benchmark performance, but also the temporal stability of localization in difficult sequences.

Efficiency Analysis. A natural concern is whether evolution-aware temporal modeling introduces extra inference overhead. To examine this, Table 7 compares representative category-agnostic trackers

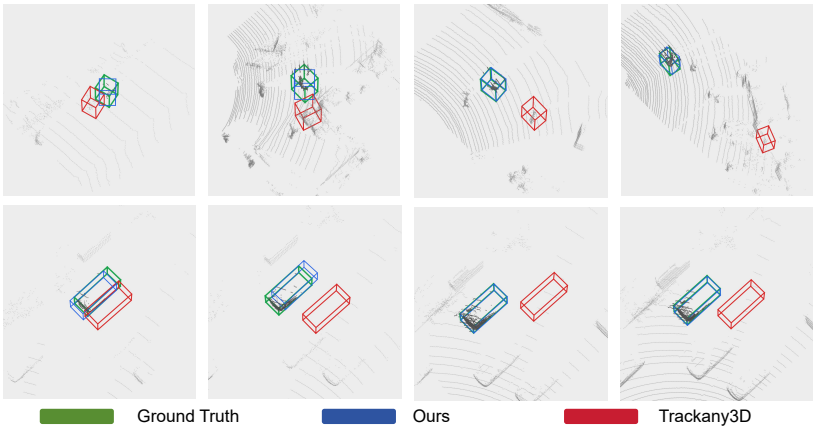

Figure 4: Visual comparison on some sequences.

in terms of average benchmark performance, FPS, and GMACs. TETrack3D achieves the best mean performance on all three datasets, with 70.2 / 87.9 on KITTI, 56.6 / 69.7 on nuScenes, and 72.2 / 78.6 on WOD. Compared with TrackAny3D [26], TETrack3D improves the three benchmarks by 3.1% / 2.5%, 2.0% / 3.4%, and 8.2% / 5.3%, while also running faster with lower computation, 104.01 FPS and 6.22 GMACs vs. 39.00 FPS and 7.32 GMACs. Although M^2Track is the fastest and lightest model, its accuracy is much lower. These results show that TETrack3D achieves a favorable balance between accuracy and efficiency, since the two temporal supervision modules are training-only and the inference model retains only a compact historical memory.

Table 7: Performance and efficiency comparison. Results are reported as Success / Precision.

Method	KITTI	nuScenes	WOD	FPS	GMACs
TrackAny3D [26]	67.1 / 85.4	54.6 / 66.3	64.0 / 73.3	39.00	7.32
CXTrack [30]	57.2 / 75.9	41.7 / 50.7	52.8 / 60.7	71.81	4.19
M^2Track [36]	61.1 / 80.9	45.0 / 55.8	61.1 / 69.3	297.85	2.54
MBPTrack [31]	56.1 / 74.9	51.1 / 61.5	58.2 / 67.7	63.39	4.13
TETrack3D	**70.2 / 87.9**	**56.6 / 69.7**	**72.2 / 78.6**	104.01	6.22

5 Conclusion

In this paper, we presented TETrack3D, a unified framework for category-agnostic 3D point cloud tracking that shifts temporal modeling from auxiliary frame-wise cues to explicit target evolution learning. TETrack3D consists of three complementary designs: historical feature reuse for preserving fine-grained historical target evidence, state evolution supervision for enforcing temporally coherent representations through future-state prediction, and temporal distribution alignment for stabilizing target-related feature distributions across frames via optimal transport. The latter two are only used during training, so inference remains efficient. Extensive experiments on KITTI, nuScenes, and WOD demonstrate that TETrack3D consistently improves over strong category-agnostic baselines and remains competitive with leading category-specific trackers, validating the effectiveness of evolution-aware temporal learning for category-agnostic 3D point cloud tracking.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. U24A20330, 6236116670, and 62406135; the Natural Science Foundation of Jiangsu Province under Grant No. BK20241198; the Gusu Innovation and Entrepreneur Leading Talents Program under Grant No. ZXL2024362; and the Fund of Key Laboratory of Education Blockchain and Intelligent Technology (Guangxi Normal University), Ministry of Education, under Grant No. EBME2026B06. The authors declare no competing interests.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[2] BaiChen Fan, Sifan Zhou, Jian Li, Shibo Zhao, Muqing Cao, and Qin Wang. Beyond frame-wise tracking: A trajectory-based paradigm for efficient point cloud tracking. *IEEE Robotics and Automation Letters*, 2026.

[3] 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*, pages 3354–3361. IEEE, 2012.

[4] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1359–1368, 2019.

[5] Xiantao Hu, Ying Tai, Xu Zhao, Chen Zhao, Zhenyu Zhang, Jun Li, Bineng Zhong, and Jian Yang. Exploiting multimodal spatial-temporal patterns for video object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3581–3589, 2025.

[6] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3d siamese voxel-to-bev tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34:28714–28727, 2021.

[7] 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 *European conference on computer vision*, pages 293–310. Springer, 2022.

[8] Kaihao Lan, Haobo Jiang, and Jin Xie. Temporal-aware siamese tracker: Integrate temporal context for 3d object tracking. In *Proceedings of the Asian Conference on Computer Vision*, pages 399–414, 2022.

[9] Yu Lin, Zhiheng Li, Yubo Cui, and Zheng Fang. Seqtrack3d: Exploring sequence information for robust 3d point cloud tracking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6959–6965. IEEE, 2024.

[10] Jiaming Liu, Yue Wu, Maoguo Gong, Qiguang Miao, Wenping Ma, Cai Xu, and Can Qin. M3sot: Multi-frame, multi-field, multi-space 3d single object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 3630–3638, 2024.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[12] Yuxuan Lu, Jiahao Nie, Zhiwei He, Hongjie Gu, and Xudong Lv. Voxeltrack: exploring voxel representation for 3d point cloud object tracking. *arXiv preprint arXiv:2408.02263*, 2024.

[13] Zhipeng Luo, Gongjie Zhang, Changqing Zhou, Zhonghua Wu, Qingyi Tao, Lewei Lu, and Shijian Lu. Modeling continuous motion for 3d point cloud object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 4026–4034, 2024.

[14] Zhipeng Luo, Changqing Zhou, Liang Pan, Gongjie Zhang, Tianrui Liu, Yueru Luo, Haiyu Zhao, Ziwei Liu, and Shijian Lu. Exploring point-bev fusion for 3d point cloud object tracking with transformer. *IEEE transactions on pattern analysis and machine intelligence*, 46(9):5921–5935, 2024.

[15] Teli Ma, Mengmeng Wang, Jimin Xiao, Huifeng Wu, and Yong Liu. Synchronize feature extracting and matching: A single branch framework for 3d object tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9953–9963, 2023.

[16] 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 *proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 1957–1965, 2023.

[17] Jiahao Nie, Zhiwei He, Xudong Lv, Xueyi Zhou, Dong-Kyu Chae, and Fei Xie. Towards category unification of 3d single object tracking on point clouds. *arXiv preprint arXiv:2401.11204*, 2024.

[18] Jiahao Nie, Fei Xie, Sifan Zhou, Xueyi Zhou, Dong-Kyu Chae, and Zhiwei He. P2p: Part-to-part motion cues guide a strong tracking framework for lidar point clouds. *International Journal of Computer Vision*, 133(8):5326–5342, 2025.

[19] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Space-time memory networks for video object segmentation with user guidance. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):442–455, 2020.

[20] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang 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*, pages 6329–6338, 2020.

[21] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023.

[22] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo 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)*, pages 1310–1316. IEEE, 2021.

[23] Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 4838–4846, 2024.

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[25] Hongyu Wang, Xiaotao Liu, Yifan Li, Meng Sun, Dian Yuan, and Jing Liu. Temporal adaptive rgbt tracking with modality prompt. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 5436–5444, 2024.

[26] Mengmeng Wang, Haonan Wang, Yulong Li, Xiangjie Kong, Jiaxin Du, Guojiang Shen, and Feng Xia. Trackany3d: Transferring pretrained 3d models for category-unified 3d point cloud tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28249–28259, 2025.

[27] Qiangqiang Wu, Yan Xia, Jia Wan, and Antoni B Chan. Boosting 3d single object tracking with 2d matching distillation and 3d pre-training. In *European Conference on Computer Vision*, pages 270–288. Springer, 2024.

[28] Qiao Wu, Kun Sun, Pei An, Mathieu Salzmann, Yanning Zhang, and Jiaqi Yang. 3d single-object tracking in point clouds with high temporal variation. In *European Conference on Computer Vision*, pages 279–296. Springer, 2024.

[29] Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19300–19309, 2024.

[30] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Cxtrack: Improving 3d point cloud tracking with contextual information. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1084–1093, 2023.

[31] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9911–9920, 2023.

[32] Weisheng Xu, Sifan Zhou, Jiaqi Xiong, Ziyu Zhao, and Zhihang Yuan. Pillartrack: Boosting pillar representation for transformer-based 3d single object tracking on point clouds. *arXiv preprint arXiv:2404.07495*, 2024.

[33] Zhefan Xu, Xiaoyang Zhan, Yumeng Xiu, Christopher Suzuki, and Kenji Shimada. Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera. *IEEE Robotics and Automation Letters*, 9(1):651–658, 2023.

[34] Jingwen Zhang, Zikun Zhou, Guangming Lu, Jiandong Tian, and Wenjie Pei. Robust 3d tracking with quality-aware shape completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 7160–7168, 2024.

[35] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13199–13208, 2021.

[36] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8111–8120, 2022.

[37] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 7588–7596, 2024.

[38] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8531–8540, 2022.

[39] Sifan Zhou, Yichao Cao, Jiahao Nie, Yuqian Fu, Ziyu Zhao, Xiaobo Lu, and Shuo Wang. Comptrack: Information bottleneck-guided low-rank dynamic token compression for point cloud tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 13773–13781, 2026.

[40] Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8120–8132, 2025.

A Technical appendices and supplementary material

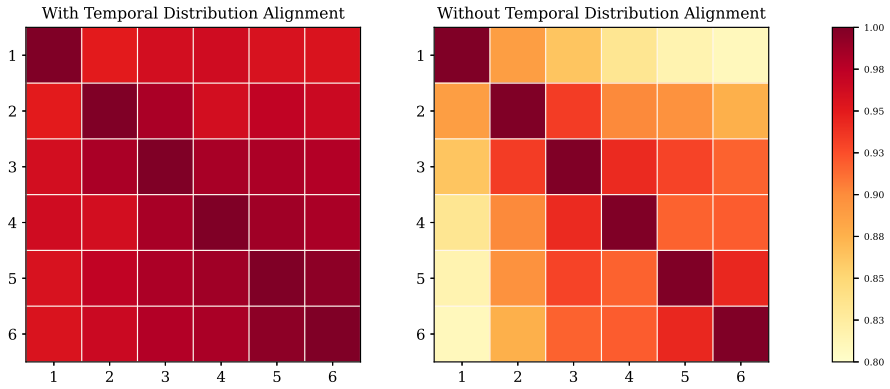

Figure 5: Foreground token similarity across six consecutive frames. Temporal distribution alignment yields a more coherent high-similarity structure than the variant without it.

Temporal Feature Consistency. A key motivation of temporal distribution alignment is to maintain stable target-related feature distributions across frames, so that the tracker can establish coherent associations under sparsity, occlusion, and viewpoint variation. To verify this, we visualize the foreground token similarity matrix across six consecutive frames in Figure 5, where each entry denotes the similarity between target-related tokens of the two corresponding frames. As shown in the figure, the variant without temporal distribution alignment produces a fragmented similarity structure, where off-diagonal entries quickly decay and indicate substantial cross-frame feature drift. In contrast, with temporal distribution alignment, the similarity matrix exhibits a uniformly high-similarity structure across all frame pairs, indicating that target-related features stay temporally consistent over a longer horizon. This validates that temporal distribution alignment effectively reduces cross-frame feature drift and stabilizes feature-level target evolution.

Future Prediction Quality vs. Tracking Accuracy. A core motivation of State Evolution Supervision is to enforce temporally coherent target dynamics through future-state prediction, so that the model can extract reliable evolution patterns from past observations. A natural question is whether this objective is truly beneficial for tracking, or merely serves as an auxiliary regularizer. If State Evolution Supervision indeed shapes the learned representations toward temporally coherent target dynamics, frames whose future states can be predicted more accurately should also be tracked more reliably. To verify this, we group all evaluation frames into eight bins (B1 to B8) according to their future-state prediction error, from low to high, and report the tracking IoU distribution within each bin in Figure 6. As the prediction error increases from B1 to B8, the mean tracking IoU drops by about 12.9%, and the IoU distribution within each bin shifts consistently toward lower values, with the lower whisker dropping noticeably as well. This monotonic trend, rather than a few isolated cases, indicates a strong correlation between future-state prediction quality and tracking accuracy. In other words, when the model can better anticipate how the target evolves over time, it also localizes the target more accurately, which suggests that the two abilities share the same underlying temporal representation. These observations validate that State Evolution Supervision does not merely act as an auxiliary training signal, but actually drives the model to learn temporally coherent target representations that translate into measurable improvements in target localization.

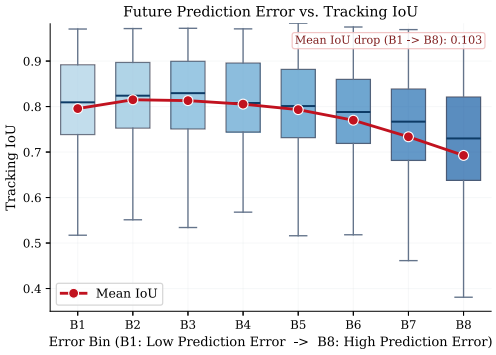

Figure 6: Future prediction error vs. tracking IoU. Lower prediction error consistently corresponds to higher tracking quality, supporting the effectiveness of State Evolution Supervision.

B Robustness to incomplete observations

B.1 Occlusion

We compare TETrack3D with TrackAny3D [26] on KITTI using the official occlusion annotations (Table 8). TETrack3D performs better in both groups, although severe occlusion reduces its tracking accuracy.

Table 8: KITTI tracking under different occlusion levels (Success / Precision).

Occlusion level	TrackAny3D	TETrack3D
Partly occluded	67.6 / 85.5	72.1 / 88.4
Largely occluded	57.9 / 77.5	63.0 / 81.9

B.2 Point-cloud sparsity

We randomly retain 25%, 50%, 75%, or 100% of each raw LiDAR frame while keeping the KITTI test sequences unchanged. Both trackers use the same fixed mask for each frame, with one mask realization per retention setting. Table 9 shows that TETrack3D maintains its advantage as observations become sparser, but both trackers degrade under severe point removal.

Table 9: KITTI tracking with different point-retention ratios (Success / Precision).

Point retention	TrackAny3D	TETrack3D
25%	55.5 / 73.5	59.5 / 76.7
50%	59.8 / 77.8	67.0 / 84.5
75%	63.8 / 82.6	68.6 / 85.9
100%	67.1 / 85.4	70.2 / 87.9

C Limitations

TETrack3D relies on historical target observations to support current localization, which introduces additional computation and memory consumption for storing and reusing historical features. Lightweight historical-memory compression is a potential direction for reducing this cost.

Severe occlusion and extreme point-cloud sparsity remain challenging despite the improvements in Tables 8 and 9. Under severe or prolonged occlusion, current observations may provide insufficient target evidence, while unreliable recent observations can weaken the historical references and allow localization errors to accumulate into tracking drift. Under extreme sparsity, the remaining target points may not preserve enough distinctive geometry to separate the target from nearby background structures. Temporal modeling alone cannot fully compensate for such missing or unreliable observations. Future work will explore confidence-aware memory updating and reference selection to limit the propagation of unreliable observations, as well as sparse-target geometry enhancement or shape completion to improve localization with limited target points.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list our contributions and scope in the abstract and in the last three paragraphs of Section introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix C discusses severe occlusion, sparse observations, and computational overhead; Appendix B provides the corresponding quantitative robustness evaluation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: In this paper, we do not present new theoretical results, as we rely on existing theories.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed network components, training procedure and training settings at Sec. Experiments, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code, configurations, and training/evaluation instructions are available at `https://github.com/NJU-PCALab/TETrack3D`. The benchmark datasets are cited in the main paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Training settings appear in Section 4.1; Appendix B specifies the additional robustness evaluation protocol, and the main-paper efficiency analysis reports inference speed and computational cost in Table 7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Tracking accuracy uses one training run per configuration, and the point-retention study uses one mask realization per setting; no error bars or confidence intervals are reported. The inference-speed results in Table 7 do not estimate tracking-accuracy variance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 specifies the GPU and CPU configuration and training epochs; Table 7 reports inference speed and computational cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read the NeurIPS Code of Ethics and find that the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive societal impacts in the Introduction.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package or dataset in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The new assets introduced in the paper are not well documented. We will improve the documentation and provide it alongside the assets in future revisions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.