Rethinking the Spatio-Temporal Alignment of End-to-End 3D Perception

Xiaoyu Li[1*], Peidong Li[2,3*], Xian Wu[1], Long Shi[1], Dedong Liu[1], Yitao Wu[1], Jiajia Fu[1], Dixiao Cui[4], Lijun Zhao[1†], Lining Sun[1]

[1]State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China
[2]IROOTECH, Guangzhou, China
[3]Wolf 1069b, Guangzhou, China
[4]Independent Researcher

Abstract

Spatio-temporal alignment is crucial for temporal modeling of end-to-end (E2E) perception in autonomous driving (AD), providing valuable structural and textural prior information. Existing methods typically rely on the attention mechanism to align objects across frames, simplifying the motion model with a unified explicit physical model (constant velocity, etc.). These approaches prefer semantic features for implicit alignment, challenging the importance of explicit motion modeling in the traditional perception paradigm. However, variations in motion states and object features across categories and frames render this alignment suboptimal. To address this, we propose HAT, a spatio-temporal alignment module that allows each object to adaptively decode the optimal alignment proposal from multiple hypotheses without direct supervision. Specifically, HAT first utilizes multiple explicit motion models to generate spatial anchors and motion-aware feature proposals for historical instances. It then performs multi-hypothesis decoding by incorporating semantic and motion cues embedded in cached object queries, ultimately providing the optimal alignment proposal for the target frame. On nuScenes, HAT consistently improves 3D temporal detectors and trackers across diverse baselines. It achieves state-of-the-art tracking results with 46.0% AMOTA on the test set when paired with the DETR3D detector. In an object-centric E2E AD method, HAT enhances perception accuracy (+1.3% mAP, +3.1% AMOTA) and reduces the collision rate by 32%. When semantics are corrupted (nuScenes-C), the enhancement of motion modeling by HAT enables more robust perception and planning in the E2E AD.

Introduction

In the autonomous driving, multi-camera 3D temporal detection (Lin et al. 2023; Liu et al. 2023; Wang et al. 2023b; Tang et al. 2025; Wang et al. 2022) and multi-object tracking (Pang et al. 2023; Zhang et al. 2022; Ding et al. 2024; Li et al. 2023b; Lin et al. 2023; Doll et al. 2023) (MOT) tasks converge through query-based data-stream. They form an integrated front-end for vision-based E2E perception systems. As illustrated in fig. 1, these methods employ a memory mechanism to store instance-level information from adjacent

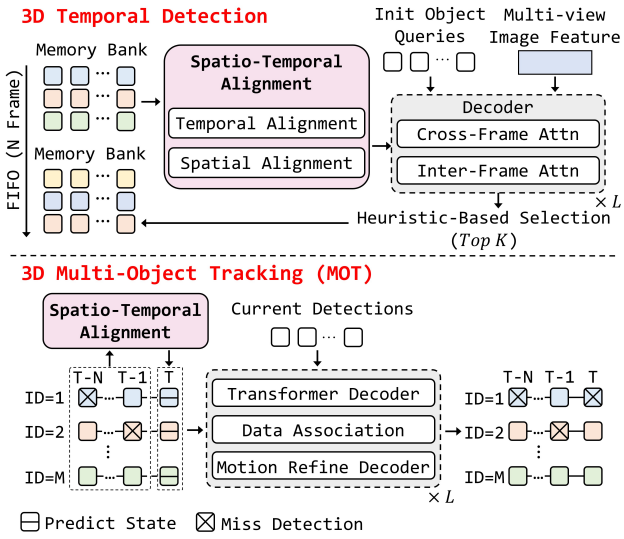

Figure 1: The STA module in detection and tracking.

frames. This cached information serves as helpful priors of the target frame, enhancing the performance of 3D perception from temporal modeling. To further bridge the representation gap across time and space, the Spatio-Temporal Alignment (STA) module is introduced, aligning the features and anchors of historical instances to the target frame.

The STA module is equally critical in modularized temporal-related methods (Ding et al. 2023; Wang et al. 2023a), typically implemented via auto-regressive filters (the Kalman Filter (Kalman 1960), etc.). Approaches like (Li et al. 2023a; Genovese 2001) improve tracking by enforcing more plausible state transitions through refined motion models. However, identifying the optimal alignment proposal for each object in these methods requires manual hyperparameter tuning and is prone to overfitting specific motion patterns. Paradoxically, recent E2E methods tend to oversimplify motion modeling, ignoring these prior advancements. These methods prefer to use query propagation to perform feature alignment and adopt a single hypothesis (constant velocity, etc.) for motion compensation, favoring inter-frame alignment in latent space (Wang et al. 2023b)

*These authors contributed equally.
†Corresponding author.

rather than in explicit motion space. This conflicts with fundamental facts that object motion is *category-specific* and *time-variant*, as revealed by (Li et al. 2023a) and fig. 3. Consequently, single-hypothesis motion modeling in current E2E perception fails to capture the diverse inter-frame transformation patterns exhibited by past instances. In contrast, we argue that the propagated query contains valuable but underutilized motion cues. These cues can be leveraged to distinguish and construct the most appropriate structural and textural prior information for corresponding objects. These observations motivate a central question: ***How can E2E perception integrate the strengths of advanced STA modules of stacked perception without inheriting their brittleness?***

To this end, we propose HAT, a multiple **H**ypotheses sp**A**tio-**T**emporal alignment module that adaptively decodes the optimal alignment proposal for each object. Specifically, we first utilize multiple predefined explicit motion models to independently generate spatially compensated anchors and motion-aware feature proposals for historical instances. This multi-hypothesis generator increases the diversity of alignment proposals, effectively modeling multiple possible states of an object while ensuring performance limits. Inspired by (Liu et al. 2023; Gao et al. 2022), we then introduce an adaptive decoder that decodes these hypotheses with dynamic weights derived from the queries. The decoder treats historical instances as guides and generates proposals as sample points in the 3D motion space. Finally, the fused anchors and motion-aware features are mixed through a motion refinement layer and a Feed-Forward Network (FFN) to enhance their representations. The refined anchors and features are treated as optimal spatio-temporal alignment proposals and sent to the corresponding task head.

On the large-scale dataset nuScenes (Caesar et al. 2020), we integrate HAT into various multi-view 3D temporal detectors and 3D trackers. Without direct supervision, HAT demonstrates strong cross-task generalization and effectiveness. On the validation set, it yields average gains of +0.7% NDS and +0.6% mAP for detection, and +1.3% MOTA and +1.0% AMOTA for tracking. Combining with ADA-Track (Ding et al. 2024), **we achieve state-of-the-art on the test set with 46.0% AMOTA** among trackers paired with DETR3D (Wang et al. 2022). Furthermore, HAT positively impacts system-level performance by improving 3D perception and reducing the collision rate in several object-centric E2E AD methods. Its robustness is further verified on the challenging nuScenes-C benchmark (Dong et al. 2023), which introduces deliberate semantic corruptions. Our extensive experiments demonstrate that motion modeling remains a crucial component in E2E 3D perception, alongside semantic cues. The primary contributions of this work are:

- We propose HAT, a plug-and-play spatio-temporal alignment module that can be seamlessly integrated into various object-centric methods in the E2E AD system.
- HAT introduces a novel explicit-implicit mixing alignment module that enhances the robustness of the E2E AD system, while overcoming the manual intervention inherent in the stacked AD system.
- HAT achieves promising improvements when integrated into existing query-based 3D temporal detectors, trackers and even E2E AD methods, serving prior information.

Related Work

Multi-Camera 3D Temporal Detection initially uses Bird's-Eye-View (BEV) features (Li et al. 2025) for temporal modeling. Recent methods instead adopt object-centric temporal propagation to reduce computational overhead. These methods typically comprise memory bank, STA module, and transformer decoder. The memory bank retains the K most confident instances from recurrent (Tang et al. 2025) or adjacent frames (Wang et al. 2023b). The STA module warps historical queries and anchors to the current frame, mitigating inter-frame discrepancies. The transformer decoder then performs cross-/inter-frame attention to retrieve current instances from the aligned past and image features (Zhu et al. 2020). Under identity-agnostic supervision, temporal detectors surprisingly capture object-level temporal consistency, driven by the query-centric attention that implicitly links similar propagated queries across frames.

Multi-Camera 3D MOT provides an explicit E2E pipeline for training detectors and trackers through collaborative query propagation. By modeling the consistency of instances, these trackers act as temporal-enhanced wrappers for off-the-shelf detectors (Liu et al. 2022; Wang et al. 2022). These methods typically follow the Tracking-By-Attention framework, where tracklet queries are updated from detection and image queries via attention. Pioneering methods (Zhang et al. 2022; Pang et al. 2023) propagate tracklet queries while simultaneously initializing detection queries within the detection decoder. During propagation, the STA module provides motion priors and semantic features for each tracklet, which are crucial for building and solving attention affinities. Recent trackers (Li et al. 2023b; Ding et al. 2024) improve recall by learning an attention map between predicted tracklets and current observations. With identity constraints, these methods achieve enhanced tracking robustness and exhibit strong detector-agnostic generalization.

Spatial-Temporal Alignment functions as a prior estimator, establishing a link between past and current reasoning. Modular tracking methods (Li et al. 2023a, 2024; Genovese 2001) reduce state–observation drift in auto-regressive filters by applying STA within motion space, typically implemented through refined motion models. However, selecting suitable models based on handcrafted rules (Li et al. 2023a) or covariance (Genovese 2001) restricts their adaptability. In contrast, E2E perception simplifies alignment by transferring structured priors through a unified physical model (Lin et al. 2023; Ding et al. 2024). These methods prefer to leverage semantic cues to project cached features across frames in latent space, enabling data-driven temporal transitions. Object motion and frame transformations can be incorporated as prompts (Wang et al. 2023b; Doll et al. 2023) to enhance temporal reasoning. Nevertheless, these approaches remain suboptimal, as a single model cannot capture the diverse motion patterns that vary across object categories and dynamic scenes. HAT addresses this limitation by jointly modeling motion and semantics. It leverages multiple explicit motion

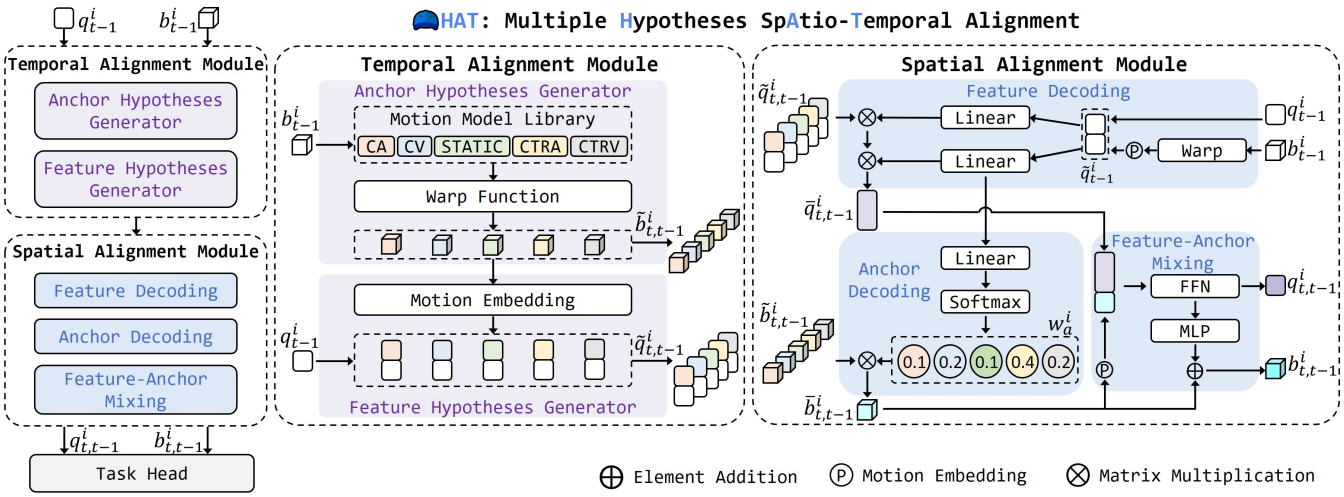

Figure 2: Architecture of HAT. HAT takes the posterior from the previous frame as input, and outputs the prior structured (anchors) and semantic (features) information for the target frame. We present the alignment process for the i-th instance.

models to generate complementary proposals, while implicitly decoding motion and semantic cues embedded within cached instances to achieve robust and adaptive alignment.

Our Approach

Problem Definition

The STA module aligns historical instances in the memory bank with the target timestamp. Without loss of generality, we consider frame $t-1$ as the historical frame and frame t as the target frame. Given the set of 3D structural anchors $B_{t-1} = \left\{b^i_{t-1}, i = 1 \dots K\right\}$ and queries $Q_{t-1} = \left\{q^i_{t-1}, i = 1 \dots K\right\}$ for the historical instances at frame $t-1$, K denotes the number of instances stored. STA propagates these instances to frame t, formulated as:

$$B_{t,t-1}, Q_{t,t-1} \longleftarrow \text{STA}\left(B_{t-1}, Q_{t-1}, \Delta t, E^t_{t-1}\right), \quad (1)$$

where $B_{t,t-1}$ and $Q_{t,t-1}$ are the spatio-temporal aligned anchors and queries. Δt is unit time interval between adjacent frames. $E^t_{t-1} = [R^t_{t-1}|T^t_{t-1}] \in \mathbb{R}^{3\times4}$ is the ego pose transformation matrix from frame $t-1$ to frame t. $R^t_{t-1} \in \mathbb{R}^{3\times3}$ and $T^t_{t-1} \in \mathbb{R}^{3\times1}$ are the rotation and translation.

An anchor b^i and its feature $q^i \in \mathbb{R}^{1\times C}$ form the i-th instance. b^i is represented as a vector $[P, D, \Theta, V] \in \mathbb{R}^{1\times10}$, where $P = [x, y, z]$ denotes the 3D center. $D = [w, l, h]$ denotes 3D size. $\Theta = [\cos\theta, \sin\theta]$ is yaw vector, where θ is the overall yaw. $V = [vx, vy]$ is the velocity vector.

Overall Architecture

The alignment is conducted in two stages. The Temporal Alignment Module first produces multiple motion-aware hypotheses informed by historical states and explicit motion models. Subsequently, the Spatial Alignment Module decodes these hypotheses using the motion cues embedded in the query, selecting the optimally aligned anchor and associated feature for propagation to the current frame as prior.

Temporal Alignment Module

This module generates multiple anchor and feature alignment proposals based on several explicit motion models. On the one hand, we employ modularized approaches (Genovese 2001) to enhance robust performance, and on the other hand, we utilize query propagation to improve adaptability.

Multiple Anchor Hypotheses Generator. Random frame sampling (Tang et al. 2025), latent ID consistency modeling (Sun et al. 2024), and lack of map collectively restrict the generalizability of fixed-horizon motion forecasting (Shi et al. 2022) in the STA module. Consequently, most methods rely on explicit motion models for anchor alignment, typically assuming a single motion hypothesis. In contrast, we define and store a set of well-established motion models in the Motion Model Library (MML). Specifically, M models are included: *Constant Velocity (CV)*, *STATIC*, *Constant Acceleration (CA)*, *Constant Turn Rate and Velocity (CTRV)*, *Constant Turn Rate and Acceleration (CTRA)*. These representative models, widely adopted in transportation scenarios (Schubert, Richter, and Wanielik 2008), capture a broad spectrum of motion patterns commonly observed in perception tasks, including stationary (*STATIC*), linear motion (*CV, CA*), and turning (*CTRV, CTRA*). These models efficiently extrapolate anchor hypotheses using Δt and the historical anchors B_{t-1}, uniformly formulated as:

$$\hat{s}_{t,t-1} = s_{t-1} + \int_{(t-1)\Delta t}^{t\Delta t} \dot{s}(\tau)d\tau = s_{t-1} + \Delta s, \quad (2)$$

where s_{t-1} and $\dot{s}$ are the state and its derivatives with respect to time in each anchor $b_{t-1} \in B_{t-1}$. Specifically, these pairs correspond to: position and velocity, velocity and acceleration, yaw and yaw rate. Notably, acceleration and yaw rate are unobservable in existing methods. Inspired by the human pose estimation task (Zeng et al. 2022), we use a multi-layer perception (MLP) to decode these states from

the instance feature q_{t-1}. Our approach to training acceleration is grounded in system observability, a principle ensuring that unobserved states (acceleration) can be inferred from measured states (position and velocity). $\hat{s}_{t,t-1}$ is the object-motion compensation state, expressed in the $t-1$ ego reference frame. Δs represents the incremental changes.

Multiple anchor hypotheses are derived from the model-specific Δs. For example, for the position x, Δs is 0 and $vx \times \Delta t$ in *STATIC* and *CV*. We perform the transformation of the m-th motion model for all historical instances B_{t-1}, obtaining $\hat{B}^m_{t,t-1}$. A linear reference frame transformation is then applied to each instance $\hat{b}^m_{t,t-1} \in \hat{B}^m_{t,t-1}$, aligning these anchors as multiple candidate priors for frame t:

$$(\tilde{b}^m_{t,t-1})^T = \mathbf{R}^t_{t-1}(\hat{b}^m_{t,t-1})^T + \mathbf{T}^t_{t-1}, \tag{3}$$

where,

$$\mathbf{R}^t_{t-1} = \mathrm{Diag}(R^t_{t-1}, I_{3\times 3}, R^t_{t-1}[:2,:2], R^t_{t-1}[:2,:2]), \tag{4}$$

$$\mathbf{T}^t_{t-1} = \left[(T^t_{t-1})^T, O_{1\times 7}\right]^T, \tag{5}$$

where $\mathrm{Diag}(\cdot)$ is the diagonalize matrix operation. $\mathbf{R}^t_{t-1} \in \mathbb{R}^{10\times 10}$ and $\mathbf{T}^t_{t-1} \in \mathbb{R}^{10\times 1}$ are augmented extrinsic matrices. O and I are the zero and identity matrix. Correspondingly, $\tilde{B}^m_{t,t-1} \in \mathbb{R}^{K\times 10}$ is defined as the set of $\tilde{b}^m_{t,t-1}$. We stack $\tilde{B}^m_{t,t-1}$ of every model, and generate the multiple anchor hypotheses as $\tilde{B}_{t,t-1} \in \mathbb{R}^{K\times M\times 10}$.

Multiple Feature Hypotheses Generator. Traditional approaches manually select the optimal anchor proposal from $\tilde{B}_{t,t-1}$. In contrast, we generate motion embeddings for each proposal, allowing the network to adaptively select the most appropriate anchor hypothesis. Given that the primary variations in $\tilde{B}_{t,t-1}$ occur in position, velocity, and yaw, we utilize the state-decoupled encoder in (Lin et al. 2023) to extract motion embeddings from $\tilde{B}_{t,t-1}$, formulated as:

$$\tilde{Q}'_{t,t-1} = \mathrm{Cat}(\Phi_P(\tilde{P}), \Phi_D(\tilde{D}), \Phi_\Theta(\tilde{\Theta}), \Phi_V(\tilde{V})), \tag{6}$$

where $\mathrm{Cat}(\cdot)$ denotes concatenation. Φ_P, Φ_D, Φ_Θ, and Φ_V are MLPs that encode the distinct motion states: center hypotheses $\tilde{P}$, size hypotheses $\tilde{D}$, yaw hypotheses $\tilde{\Theta}$, and velocity hypotheses $\tilde{V}$, respectively. Motion embedding $\tilde{Q}'_{t,t-1} \in \mathbb{R}^{K\times M\times C}$ captures information from a single frame. We then concatenate $\tilde{Q}'_{t,t-1}$ with the propagated query Q_{t-1} to incorporate temporal object characteristics:

$$\tilde{Q}_{t,t-1} = \mathrm{Cat}(\tilde{Q}'_{t,t-1}, Q_{t-1}), \tag{7}$$

where $\tilde{Q}_{t,t-1} \in \mathbb{R}^{K\times M\times 2C}$ denotes multiple motion-aware feature hypotheses corresponding to $\tilde{B}_{t,t-1}$. $\tilde{Q}_{t,t-1}$ highlights inter-hypothesis variability while capturing the consistency between historical instances and each hypothesis.

Spatial Alignment Module

The STA module in the existing methods follows the Single-Input, Single-Output (SISO) rule. In this section, we decode the single optimal hypothesis from $\tilde{B}_{t,t-1}$ and $\tilde{Q}_{t,t-1}$. A key insight that we treat $\tilde{B}_{t,t-1}$ and $\tilde{Q}_{t,t-1}$ as the sampling points and features for historical instances in 3D motion space, analogous to the 2D sampling mechanism. Inspired by (Liu et al. 2023; Gao et al. 2022), we introduce an adaptive multiple hypotheses decoder that leverages dynamic weights derived from the propagated instances. This decoder consists of three key components: feature decoding, anchor decoding, and feature-anchor mixing.

Feature Decoding. Following (Liu et al. 2023; Gao et al. 2022), we first generate dynamic weights from B_{t-1} and Q_{t-1}, which subsequently guide the fusion of multiple features. B_{t-1} is directly warped to frame t through eq. (3), resulting in $\tilde{B}_{t-1}$. Via eq. (6) and eq. (7), $\tilde{B}_{t-1}$ is encoded as motion embedding, and concatenated with Q_{t-1}, yielding $\tilde{Q}_{t-1} \in \mathbb{R}^{K\times 2C}$. We apply two linear layers $\mathrm{L}(\cdot)$ on $\tilde{Q}_{t-1}$ to obtain the dynamic weights, is described as:

$$W_c = \mathrm{L}_c(\tilde{Q}_{t-1}), W_f = \mathrm{L}_f(\tilde{Q}_{t-1}), \tag{8}$$

where the parameters of $\mathrm{L}_c(\cdot)$ and $\mathrm{L}_f(\cdot)$ are shared across instances. $W_c \in \mathbb{R}^{K\times 2C\times 2C}$ and $W_f \in \mathbb{R}^{K\times M\times 1}$ are the weights of channel fusion and feature hypotheses fusion. An MLP-like architecture then uses W_c and W_f to fuse $\tilde{Q}_{t,t-1}$:

$$\begin{aligned} \acute{Q}_{t,t-1} &= \sigma(\mathrm{LN}(\tilde{Q}_{t,t-1} \otimes W_c)), \\ \bar{Q}_{t,t-1} &= \sigma(\mathrm{LN}(W_f \otimes \acute{Q}_{t,t-1})), \end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is the activation function, and $\mathrm{LN}(\cdot)$ represents layer normalization. The operator $\otimes$ denotes the generalized matrix multiplication, including dimension-aligned operations. $\bar{Q}_{t,t-1} \in \mathbb{R}^{K\times 2C}$ is the motion-aware feature decoded from various feature hypotheses. This feature selectively integrates motion embeddings from distinct motion models, guided by the motion cue in the propagated queries.

Anchor Decoding. Unlike the 2D sampling mechanism (Gao et al. 2022; Liu et al. 2023), which focuses solely on sampling features, the STA module necessitates decoding and propagating sampling points (multiple anchor hypotheses) to the target frame. Inspired by posterior estimation of Interacting Multiple Model (IMM) filter (Genovese 2001), we implement a weighted sum approach to decode $\tilde{B}_{t,t-1}$.

In contrast to manually setting switching likelihood (Genovese 2001), we leverage the network to regress the weights adaptively. Specifically, we perform a linear transformation and softmax function on the feature hypotheses weight W_f to obtain the final weight. This can be characterized as:

$$W_a = \mathrm{Softmax}(\mathrm{L}_a(W_f)), \tag{10}$$

where L_a is the linear layer. The parameters are shared across instances. $W_a \in \mathbb{R}^{K\times M\times 1}$ is the anchor hypotheses weight. We then use W_a to perform a weighted sum:

$$\bar{B}_{t,t-1} = W_a \otimes \tilde{B}_{t,t-1}, \tag{11}$$

where $\bar{B}_{t,t-1} \in \mathbb{R}^{K\times 10}$ is the optimal anchor proposal for adaptive decoding from $\tilde{B}_{t,t-1}$. Although the anchor motion models in MML are explicit and fixed, our implicit decoder improves the ability to fit diverse motions.

	Stage	Method	Detection		Tracking		Planning	
			mAP↑	NDS↑	AMOTA↑	MOTA↑	L2 (m)↓	CR (%) ↓
nuScenes	1st	SparseDrive†	41.9	53.0	38.2	35.5	-	-
		SparseDrive-HAT	**42.1** *(+0.2)*	**53.1** *(+0.1)*	**40.0** *(+1.8)*	**37.2** *(+1.7)*	-	-
	2nd	SparseDrive†	41.2	52.2	36.9	34.2	0.63	0.123
		SparseDrive-HAT	**42.5** *(+1.3)*	**53.1** *(+0.9)*	**40.0** *(+3.1)*	**36.7** *(+2.5)*	**0.60** *(-0.03)*	**0.084** *(-32%)*
		DiffusionDrive	41.2	52.2	37.5	34.8	**0.57**	0.080
		DiffusionDrive-HAT	**42.7** *(+1.5)*	**54.0** *(+1.8)*	**40.2** *(+2.7)*	**36.7** *(+1.9)*	0.58	**0.042** *(-48%)*
	-	SSR (Li and Cui 2025)	-				0.39	0.06
nuScenes-C	2nd w/ Snow	SparseDrive†	18.9	34.1	13.1	14.1	0.74	0.156
		SparseDrive-HAT	**23.1** *(+4.2)*	**39.1** *(+5.0)*	**18.0** *(+4.9)*	**17.4** *(+3.3)*	**0.737** *(-0.003)*	**0.122** *(-22%)*
	2nd w/ Fog	SparseDrive†	**37.2**	49.6	32.6	30.1	**0.61**	0.108
		SparseDrive-HAT	37.1	**50.3** *(+0.7)*	**34.3** *(+1.7)*	**31.7** *(+1.6)*	0.63	**0.078** *(-28%)*

Table 1: Comparison on nuScenes and nuScenes-C validation sets for E2E AD task. †: Our reproduced results by official code.

Feature-Anchor Mixing. We perform enhancements on $\bar{Q}_{t,t-1}$ and $\bar{B}_{t,t-1}$, ultimately producing $Q_{t,t-1}$ and $B_{t,t-1}$ for frame t. For the feature aspect, via eq. (6), we encode $\bar{B}_{t,t-1}$ into motion embedding. This embedding is then concatenated with $\bar{Q}_{t,t-1}$ to obtain the enhanced motion-aware feature. An FFN subsequently compresses the dimensions of the enhanced features to obtain the final output $Q_{t,t-1}$. For the anchor aspect, we apply an MLP on $Q_{t,t-1}$, refining the decoded anchors $\bar{B}_{t,t-1}$ to improve quality:

$$B_{t,t-1} = \bar{B}_{t,t-1} + \Phi_r(Q_{t,t-1}), \tag{12}$$

where Φ_r is MLP for motion refinement. $B_{t,t-1}$ and $Q_{t,t-1}$ represent the optimal anchor and feature alignment for frame t, which are subsequently forwarded to the task head.

Stability Analysis

HAT maintains baseline accuracy without direct supervision. The adaptive increment of aligned position $\bar{X}_{t,t-1}$ over the warped historical anchor is constrained within $\big(R^t_{t-1}\min(\Delta X^m),\ R^t_{t-1}\max(\Delta X^m)\big)$. ΔX^m is the m-th model compensation. As these models are physically grounded, the constraint stabilizes $\bar{X}_{t,t-1}$.

Experiment

Dataset and Implementation Details

Dataset and Metrics. We evaluate HAT on the **nuScenes** dataset, which includes 1K scenes with 1.4M 3D annotations across multiple categories. It provides data from 6 cameras, 1 LiDAR, along with ego pose. HAT is further evaluated on **nuScenes-C** (Dong et al. 2023), which introduces visual corruption under diverse weather conditions. For **detection**, we adopt nuScenes Detection Score (NDS) and mean Average Precision (mAP) as primary metrics, and report Average Translation/Orientation/Velocity Error (ATE, AOE, AVE). For **tracking**, we report Average MOT Accuracy/Precision (AMOTA, AMOTP), MOT Accuracy (MOTA), along with ID Switch (IDS), False Positives/Negatives (FP, FN). For **E2E AD**, we report L2 error and Collision Rate (CR).

Implementation Details. For fair comparison, HAT is integrated into open-source baselines with identical configurations. To thoroughly assess its generalizability within the E2E pipeline, we integrate HAT into SparseDrive (Sun et al. 2024) and DiffusionDrive (Liao et al. 2025). In independent 3D perception tasks, we apply HAT to several temporal detectors (StreamPETR (Wang et al. 2023b), Sparse4D (Lin et al. 2023), SimPB (Tang et al. 2025)) and 3D MOT methods (ADA-Track (Ding et al. 2024), StreamPETR). Image resolutions and backbone settings are listed in table 3 and table 2. No pretraining or direct supervision is used for HAT. In StreamPETR, HAT fully replaces MLN. SmoothNet (Zeng et al. 2022) is employed to regress acceleration and yaw rate in an unsupervised manner, with outputs constrained to ± 0.1. Motion embeddings for anchor hypotheses are extracted via the anchor encoder from (Lin et al. 2023).

Comparative Evaluation

E2E Autonomous Driving. In table 1, we apply HAT to a query-based E2E AD method, SparseDrive. HAT improves the perception performance of the entire system. For detection, HAT improves the baseline with 1.3% mAP and 0.9% NDS. For tracking, a larger 3.1% AMOTA and a larger 2.5% MOTA are introduced by integrating HAT. Meanwhile, HAT reduces the trajectory error of both ego and objects, providing a lower CR (-32%) to ensure safety. Integrating HAT into DiffusionDrive (Liao et al. 2025) demonstrates its stability under stronger baselines. We also surprisingly find that integrating **HAT can prevent E2E AD from the usual decrease of perception performance when joint-training with motion & planning tasks in the second stage**. Furthermore, on nuScenes-C under semantic corruption (snow), the motion cues enhanced by HAT significantly improve the robustness of perception within the E2E AD framework. HAT yields improvements of +5.0% NDS and +4.9% AMOTA in detection and tracking, while reducing the collision rate by 22%. In terms of computational overhead, HAT introduces an additional 7ms latency per frame over the 111ms baseline, indicating practical deployability.

Set	Tracker	E2E	Detector	Backbone	MOTA↑	AMOTA↑	FP↓	FN↓	IDS↓	AMOTP↓
Val	ADA-Track	✓	DETR3D	R101	34.7	38.4	14358	**38035**	839	1.378
	ADA-Track-HAT				**36.4** *(+1.7)*	**39.7** *(+1.3)*	**13100**	38121	**752**	**1.344**
	StreamPETR	×	StreamPETR	V2-99	46.1	52.6	**12594**	33380	**742**	1.129
	StreamPETR-HAT				**47.0** *(+0.9)*	**53.3** *(+0.7)*	14073	**30946**	775	**1.107**
Test	MUTR3D	✓	DETR3D	V2-99	24.5	27.0	15372	56874	6018	1.494
	PF-Track				37.8	43.4	19048	42758	**249**	1.252
	STAR-Track				40.6	43.9	–	–	607	1.256
	ADA-Track	✓	DETR3D	V2-99	40.6	45.6	15699	**39680**	834	1.237
	ADA-Track-HAT†				**41.6** *(+1.0)*	**46.0** *(+0.4)*	**15235**	39799	850	**1.236**

Table 2: Comparison results on nuScenes validation and test sets for the tracking benchmark. E2E: end-to-end trackers that integrate detection and tracking through object queries. †: we re-implement the baseline as it is closed-source on the test set.

Detector	NDS↑	mAP↑	mATE↓	mAOE↓	mAVE↓
StreamPETR	57.1	48.2	0.61	0.38	0.26
w/ HAT	**57.8** *(+0.7)*	**48.7** *(+0.5)*	**0.59**	**0.37**	**0.24**
Sparse4D	56.4	46.5	0.54	0.46	0.22
w/ HAT	**57.3** *(+0.9)*	**47.0** *(+0.5)*	**0.53**	**0.42**	**0.21**
SimPB	58.6	47.9	**0.54**	**0.32**	0.22
w/ HAT	**59.0** *(+0.4)*	**48.8** *(+0.9)*	0.55	0.33	**0.21**

Table 3: Comparison results on nuScenes validation set for the detection benchmark. All methods use an input image resolution of 256 × 704. StreamPETR employs V2-99 (Lee and Park 2020) as backbone, others use R50 (He et al. 2016).

Method	NDS↑	mAP↑	mATE↓	mAOE↓	mAVE↓
w/o MLN*	57.0	48.1	0.614	0.381	0.257
MLN (Wang et al. 2023b)	57.1	48.2	0.610	0.375	0.263
LMM† (Doll et al. 2023)	57.5	48.5	0.611	**0.367**	**0.185**
HAT (Ours)	**57.8**	**48.7**	**0.593**	0.374	0.244

Table 4: Distinct STA modules in StreamPETR. †: the use of pretraining scheme in trajectory prediction. *: we disable MLN and reproduce the results.

Detection. As shown in table 3, HAT enhances various 3D temporal detectors on the nuScenes validation set, demonstrating generalization and effectiveness. It consistently improves advanced query-based detectors, increasing NDS and mAP by (0.7%, 0.9%, 0.4%) and (0.5%, 0.5%, 0.9%) for StreamPETR, Sparse4D, and SimPB. Embedding HAT also reduces heading error and velocity error, which stem from combining multiple well-established motion models.

Multi-Object Tracking. The table 2 demonstrates that the improvements of combining HAT in 3D MOT are also pronounced. HAT boosts the E2E tracker ADA-Track by 1.3% in AMOTA and 1.7% in MOTA. Additionally, with enhanced velocity and yaw observations, HAT improves the modularized tracker StreamPETR, yielding gains of 0.7% in AMOTA and 0.9% in MOTA. On the test set, combining HAT with ADA-Track **achieves state-of-the-art performance, with 46.0% AMOTA** among trackers paired with DETR3D detector. HAT improves the baseline across key metrics (+0.4% AMOTA, +1% MOTA), demonstrating the effectiveness. We argue that the tracking task, which typically relies on supervision of ID consistency, particularly benefits from the optimal alignment proposal decoding.

Method	*Pedestrian*			*Bicycle*		
	AP ↑	AOE ↓	AVE ↓	AP ↑	AOE ↓	AVE ↓
MLN	54.3	**0.406**	0.302	48.6	**0.655**	0.174
HAT	**55.2**	0.414	**0.289**	**50.5**	0.666	**0.139**

Table 5: Category-wise performance in StreamPETR.

Motion Model Library					NDS↑	mAP↑	mATE↓	mAOE↓	mAVE↓
CV	STATIC	CA	CTRA	CTRV					
✓	×	×	×	×	56.5	45.7	0.55	**0.41**	**0.21**
×	✓	×	×	×	56.4	46.2	0.56	0.43	0.22
✓	✓	✓	×	×	56.6	46.3	**0.53**	0.53	0.23
✓	✓	✓	✓	✓	**57.3**	**47.0**	**0.53**	0.42	**0.21**
×	×	×	×	×	55.5	45.7	0.55	0.48	0.27

Table 6: Different motion model combinations in Sparse4D.

Ablation Studies and Qualitative Analysis

Effect of the STA Module. In table 4, embedding STA module generally yields a performance boost (Line 1 vs. Lines 2–4). This reveals the gap between the past and current, necessitating mapping in the state space. HAT outperforms MLN, which only leverages semantic cues for implicit alignment, by 0.7% NDS and 0.5% mAP. This underscores the significance of motion cues in STA module. Compared to LMM, which regresses inter-frame feature projections using supervised networks, HAT employs an auto-regressive method with propagated queries. Furthermore, incorporating diverse proposals, HAT achieves improvements of 0.3% NDS and 0.2% mAP over LMM. In table 5, HAT achieves notable improvements over MLN on non-rigid classes, en-

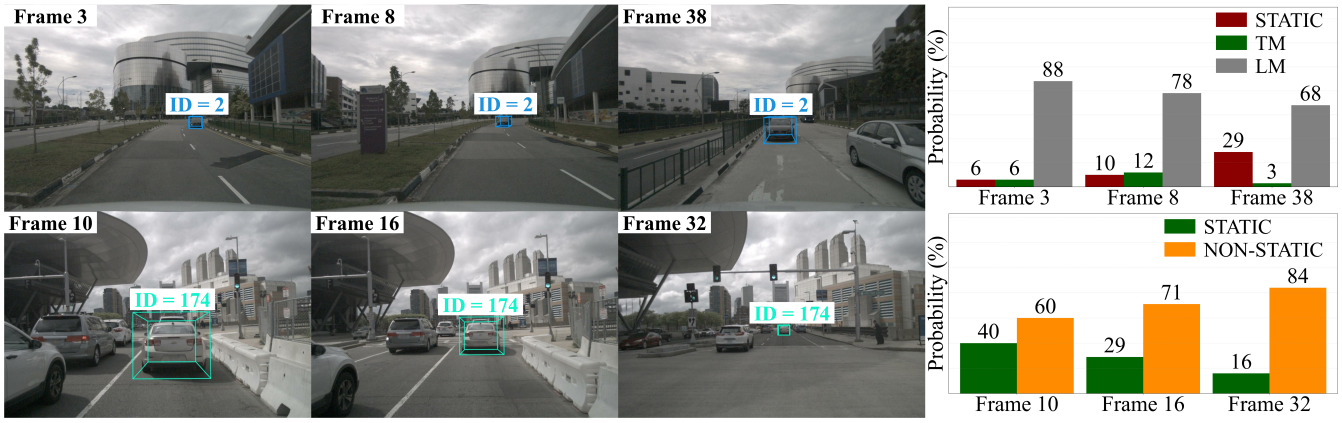

Figure 3: Visualization results of the decoding weights W_a for anchor hypotheses in ADA-Track-HAT on nuScenes. TM means Turning Motion model (*CTRV, CTRA*), LM means Linear Motion model (*CV, CA*).

Method	MOTA↑	AMOTA↑	FP↓	FN↓	IDS↓	AMOTP↓
Baseline	60.7	71.2	**13010**	19281	**341**	**0.459**
Baseline-HAT	**60.8**	71.2	13470	**19095**	384	0.523

Table 7: Integrating HAT in 3DMOTFormer.

hancing robustness in unstructured environments.

Effect of the Combination of Motion Models in MML. In table 6, increasing motion models in MML, based on Sparse4D, consistently enhances detection accuracy. Incorporating multiple hypotheses improves generalization and mitigates overfitting, yielding significant improvements in both NDS and mAP compared to a single hypothesis (Line 1-2). Compared with Line 3 without modeling the transition of yaw angle, mAOE has been greatly reduced by including *CTRV* and *CTRA* (Line 4), while the mAP is also improved by +0.7%. We further evaluate explicit motion models by using only implicit query compensation (MLN) in MML. Increased mAVE and mAOE highlight the importance of physical models for accurate motion estimation.

Effect of the Representation of Historical Instance. To investigate multi-hypothesis decoding, we integrate HAT into 3DMOTFormer (Ding et al. 2023), which propagates information via 3D decoded anchors. We first encode the recurrent anchors as motion embeddings. As shown in table 7, HAT yields marginal improvement. We attribute this to insufficient motion cues in the decoded structure, limiting effective proposal fusion. In contrast, temporally propagated queries provide rich semantic and motion cues, enabling better selection, as evidenced by gains in table 3 and table 2.

Visualization. Qualitative results demonstrate the efficacy of HAT. As shown in fig. 4, HAT reduces prior FPs by enhancing the temporal consistency of object instances, improving detection accuracy. Furthermore, fig. 3 illustrates the effectiveness of HAT in tracking by assigning greater weight to the turning-motion patterns during lane changes and to the static model during braking. These results demonstrate that HAT dynamically modulates the contribution of explicit models based on the context, enhancing alignment.

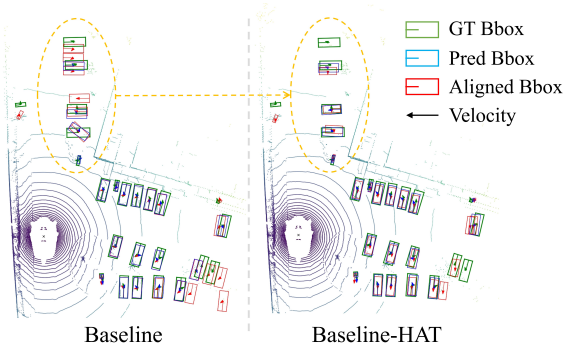

Figure 4: Visualization results of Sparse4D and Sparse4D-HAT on the BEV plane on nuScenes dataset. The ground-truth, final prediction B_t and temporal-aligned detection $B_{t,t-1}$ are depicted as green, blue, and red rectangles.

Conclusion

In this paper, we review the contribution of motion and semantic information to the STA module. The proposed HAT introduces a learnable STA module, adaptively decoding the optimal proposal from hypotheses extrapolated from multiple well-established motion models. Meanwhile, with low integration overhead, HAT exhibits impressive cross-task generalization on various AD tasks. Our extensive experiments demonstrate that motion modeling still plays a crucial role in end-to-end 3D perception, alongside semantic cues.

Limitation. Nonetheless, the multi-hypothesis decoding mechanism in HAT leverages the motion cues inherent in the temporally progressive query. As indicated in table 7, its effectiveness is reduced for methods that depend on decoded bounding boxes as the sole instance representation.

Acknowledgements

This work is supported by the Self-Planned Task of State Key Laboratory of Robotics (SKLRS202501E).

References

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 11621–11631.

Ding, S.; Rehder, E.; Schneider, L.; Cordts, M.; and Gall, J. 2023. 3dmotformer: Graph transformer for online 3d multi-object tracking. In *ICCV*, 9784–9794.

Ding, S.; Schneider, L.; Cordts, M.; and Gall, J. 2024. ADA-Track: End-to-End Multi-Camera 3D Multi-Object Tracking with Alternating Detection and Association. In *CVPR*, 15184–15194.

Doll, S.; Hanselmann, N.; Schneider, L.; Schulz, R.; Enzweiler, M.; and Lensch, H. P. 2023. STAR-Track: Latent Motion Models for End-to-End 3D Object Tracking with Adaptive Spatio-Temporal Appearance Representations. *IEEE Robotics and Automation Letters*.

Dong, Y.; Kang, C.; Zhang, J.; Zhu, Z.; Wang, Y.; Yang, X.; Su, H.; Wei, X.; and Zhu, J. 2023. Benchmarking robustness of 3d object detection to common corruptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1022–1032.

Gao, Z.; Wang, L.; Han, B.; and Guo, S. 2022. AdaMixer: A Fast-Converging Query-Based Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5364–5373.

Genovese, A. F. 2001. The interacting multiple model algorithm for accurate state estimation of maneuvering targets. *Johns Hopkins APL technical digest*, 22(4): 614–623.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems.

Lee, Y.; and Park, J. 2020. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13906–13915.

Li, P.; and Cui, D. 2025. Navigation-Guided Sparse Scene Representation for End-to-End Autonomous Driving. In *International Conference on Learning Representations (ICLR)*.

Li, X.; Liu, D.; Wu, Y.; Wu, X.; Zhao, L.; and Gao, J. 2024. Fast-poly: A fast polyhedral algorithm for 3D multi-object tracking. *IEEE Robotics and Automation Letters*.

Li, X.; Xie, T.; Liu, D.; Gao, J.; Dai, K.; Jiang, Z.; Zhao, L.; and Wang, K. 2023a. Poly-mot: A polyhedral framework for 3d multi-object tracking. In *IROS*, 9391–9398. IEEE.

Li, Y.; Yu, Z.; Philion, J.; Anandkumar, A.; Fidler, S.; Jia, J.; and Alvarez, J. 2023b. End-to-end 3d tracking with decoupled queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18302–18311.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Yu, Q.; and Dai, J. 2025. BEVFormer: Learning Bird's-Eye-View Representation From LiDAR-Camera via Spatiotemporal Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3): 2020–2036.

Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; et al. 2025. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12037–12047.

Lin, X.; Pei, Z.; Lin, T.; Huang, L.; and Su, Z. 2023. Sparse4d v3: Advancing end-to-end 3d detection and tracking. *arXiv preprint arXiv:2311.11722*.

Liu, H.; Teng, Y.; Lu, T.; Wang, H.; and Wang, L. 2023. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18580–18590.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, 531–548. Springer.

Pang, Z.; Li, J.; Tokmakov, P.; Chen, D.; Zagoruyko, S.; and Wang, Y.-X. 2023. Standing Between Past and Future: Spatio-Temporal Modeling for Multi-Camera 3D Multi-Object Tracking. In *CVPR*, 17928–17938.

Schubert, R.; Richter, E.; and Wanielik, G. 2008. Comparison and evaluation of advanced motion models for vehicle tracking. In *2008 11th international conference on information fusion*, 1–6. IEEE.

Shi, S.; Jiang, L.; Dai, D.; and Schiele, B. 2022. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35: 6531–6543.

Sun, W.; Lin, X.; Shi, Y.; Zhang, C.; Wu, H.; and Zheng, S. 2024. SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation. *arXiv preprint arXiv:2405.19620*.

Tang, Y.; Meng, Z.; Chen, G.; and Cheng, E. 2025. Simpb: A single model for 2d and 3d object detection from multiple cameras. In *European Conference on Computer Vision*, 1–17. Springer.

Wang, L.; Zhang, X.; Qin, W.; Li, X.; Gao, J.; Yang, L.; Li, Z.; Li, J.; Zhu, L.; Wang, H.; et al. 2023a. Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-lidar fusion. *T-ITS*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023b. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3621–3631.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 180–191. PMLR.

Zeng, A.; Yang, L.; Ju, X.; Li, J.; Wang, J.; and Xu, Q. 2022. Smoothnet: A plug-and-play network for refining human poses in videos. In *European Conference on Computer Vision*, 625–642. Springer.

Zhang, T.; Chen, X.; Wang, Y.; Wang, Y.; and Zhao, H. 2022. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *CVPR*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.