BeyondSight: Object Permanence for End-to-End Autonomous Driving

Sandro Papais[1], Letian Wang[1], Mudit Jain[2], Behnaz Rezaei[2], and Steven L. Waslander[1]

[1] University of Toronto
{sandro.papais, letian.wang, steven.waslander}@robotics.utias.utoronto.ca
[2] Automated Driving, Qualcomm Technologies, Inc.
{mudijain, brezaei}@qti.qualcomm.com

Project page: https://beyondsight-eccv.github.io

Abstract. Autonomous driving operates in partially observable environments where actors may become fully occluded by other vehicles or infrastructure. Most end-to-end driving systems implicitly couple actor existence to instantaneous observations, causing actor hypotheses to degrade or disappear during prolonged occlusion and removing potentially critical agents from downstream prediction and planning. We introduce **BeyondSight**, a permanence-aware end-to-end driving framework that decouples actor existence from observability by maintaining persistent actor hypotheses over time. BeyondSight propagates actor queries temporally and updates them with observation-conditioned evidence, enabling joint perception, prediction, and planning to reason about actors even when they are temporarily unobservable. To enable principled training and evaluation of persistence-aware models, we further introduce **nuScenes-Permanence**, an extension of nuScenes that provides supervision and observability-conditioned evaluation for unobservable actors. Experiments show that BeyondSight substantially improves reasoning under occlusion, increasing detection performance for unobservable actors from 0 to 0.249 mAP while reducing planning error from 0.61 to 0.54 $L2_{avg}$. These results highlight *object permanence* as an important modeling principle for robust end-to-end autonomous driving.

Keywords: Object Permanence · End-to-End Autonomous Driving · Spatiotemporal Reasoning · Partial Observability

1 Introduction

Autonomous driving operates in partially observable environments where actors frequently become fully unobservable due to occlusion or sensor limits. Our analysis of nuScenes shows that approximately 30% of actors are fully unobservable at each timestep, including many within close proximity to the ego vehicle. Therefore, safe driving requires maintaining persistent hypotheses about actors even when they are temporarily unobservable. This capability corresponds to

object permanence: the ability to reason about objects that continue to exist despite missing observations.

Recent end-to-end autonomous driving (E2E-AD) systems jointly model perception, prediction, and planning within a unified architecture. However, most existing approaches implicitly couple actor existence to instantaneous observations. When an actor becomes fully unobservable, its representation often degrades or disappears entirely (Fig. 1), removing it from the scene representation used by downstream prediction and planning modules. As a result, the planner must reason only over currently visible agents, leading to reactive or brittle behavior in occlusion-heavy scenarios such as intersections and crosswalks. While temporal aggregation improves short-term stability, it does not explicitly model actor persistence during extended observability gaps.

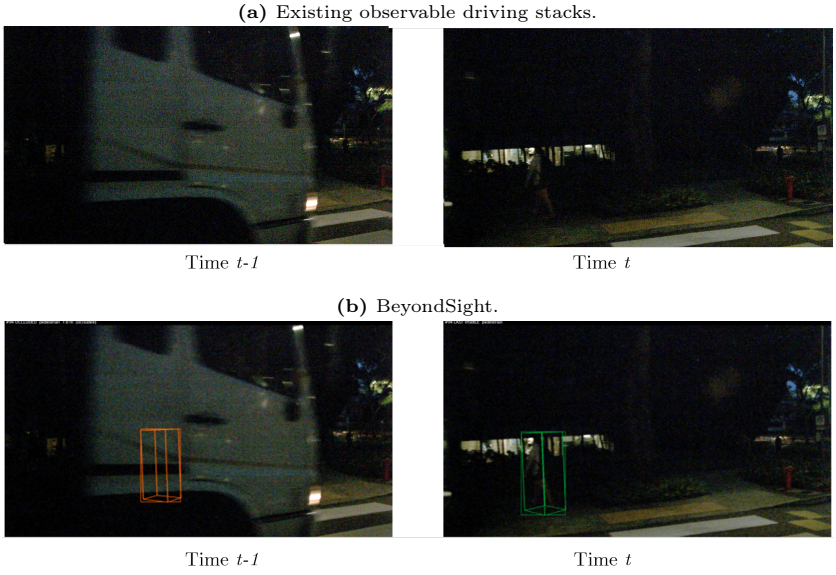

Fig. 1: Object permanence in prolonged occlusion. A pedestrian near a crosswalk becomes occluded. Top: End-to-end driving models (ex. SparseDrive) drop the actor hypothesis. Bottom: BeyondSight maintains a persistent representation during occlusion.

Current perception benchmarks implicitly equate observability with existence: when an actor becomes fully occluded, it typically disappears from both supervision and evaluation. This prevents models from learning persistent representations across observability gaps. To address this limitation, we introduce **nuScenes-Permanence**, an extension of nuScenes that provides annotations and evaluation protocols for unobservable actors, enabling systematic training and evaluation of object permanence.

Building on this benchmark, we introduce **BeyondSight**, a permanence-aware end-to-end driving framework that explicitly decouples actor existence from instantaneous observability by maintaining persistent actor hypotheses over time. BeyondSight extends sparse-query scene representations with temporal propagation and observation-conditioned updates, allowing actor hypotheses to persist through observability gaps. Propagated and observation-conditioned hypotheses are fused into a unified representation that contains both observable and unobservable actors. This persistent scene representation provides richer context for downstream motion prediction and planning, improving decision-making in occlusion-heavy scenarios.

Our contributions are summarized as follows:

- We formalize **object permanence for end-to-end autonomous driving**, defining the requirement that actors previously observed remain represented even when temporarily unobservable in both model representations and dataset annotations.
- We introduce **nuScenes-Permanence**, extending nuScenes with unobservable actor annotations and an observability-conditioned evaluation protocol that enables systematic training and evaluation of persistence-aware models.
- We propose **BeyondSight**, a permanence-aware extension to sparse-query end-to-end driving architectures that maintains persistent actor hypotheses through temporal propagation and observability-aware fusion.

Extensive experiments on the nuScenes benchmark demonstrate the effectiveness of permanence-aware reasoning. On nuScenes, BeyondSight improves planning error from 0.61 to 0.54 $L2_{avg}$ while maintaining competitive perception performance. On the proposed nuScenes-Permanence benchmark, it dramatically improves reasoning about occluded actors, increasing mAP_{unobs} from 0 to 0.249. Qualitative results further show that the model maintains stable hypotheses for actors that become fully occluded and produces safer planning behavior in complex scenes with prolonged occlusions.

2 Related Work

Vision-Based End-to-End Autonomous Driving. Vision-based end-to-end autonomous driving (E2E-AD) jointly optimizes perception, motion forecasting, and planning within a single differentiable stack. Most modern E2E-AD systems operate in BEV, combining camera-to-BEV lifting with temporal aggregation to support long-horizon planning. Lift-Splat-Shoot [13], ST-P3 [4], and BEVFormer [10] established the core recipe of BEV scene encoding and recurrent or attention-based temporal fusion. Building on this, UniAD [5] unified detection, tracking, prediction, and planning in a planning-centric architecture, while VAD [8] introduced vectorized scene representations that improve efficiency and downstream controllability. Robust motion prediction remains a central requirement for deployable autonomy, especially under uncertainty, distribution shift, and interaction-heavy driving scenarios [21].

Recent work targets scalability and latency by replacing dense BEV supervision with sparse instance-centric representations. SparseDrive [17] uses sparse queries to represent agents and map elements for joint detection, prediction, and planning. SSR [9] further prunes supervision and computation via navigation-guided sparse tokens, while DriveAdapter [6] and DriveTransformer [7] reduce coupling and improve streaming behavior through teacher–student decoupling and unified query processing. Despite architectural progress, most E2E-AD stacks still rely on training signals and metrics defined over *currently observable* actors by filtering annotations with zero LiDAR points, effectively removing fully occluded actors from supervision and evaluation. As a result, actor hypotheses often degrade or disappear after missed detections or prolonged occlusion, leading to weak temporal consistency and limited recovery once the actor reappears.

Temporal Modeling and Query Propagation. Temporal reasoning in vision-based driving is commonly implemented as (i) BEV feature recurrence or (ii) object-centric query propagation. BEVFormer [10] aggregates historical BEV features with spatiotemporal attention, improving temporal stability for BEV reasoning. In object-centric pipelines, StreamPETR [22] propagates instance queries across frames to enable efficient temporal association and multi-view 3D detection. More recent E2E-AD systems extend these ideas to multi-task temporal coherence: BridgeAD [24] leverages historical prediction to inform present perception and planning, MomAD [16] introduces momentum objectives to stabilize planning, and ForeSight [12] shares memory between detection and forecasting to improve streaming consistency. Related multi-object tracking methods also study hypothesis persistence through missed observations and association ambiguity. SWTrack [11] maintains multiple 3D tracking hypotheses in a sliding window, while SCATr [2] mitigates query suppression for joint detection and tracking.

These approaches improve short-horizon stability, association, and interaction modeling, but query survival is still largely driven by recent observations and evaluation is typically defined only where ground truth remains observable. As a result, hypotheses for fully unobservable actors often decay or disappear, yielding missing forecasts and incomplete context for planning. BeyondSight instead treats persistence as a first-class requirement: temporally propagated hypotheses remain valid independent of immediate sensor evidence, and are explicitly supervised and evaluated across observability gaps.

Occlusion and Object Permanence. Object permanence under occlusion has been studied in video understanding and tracking, where learning to preserve identity and state through occlusions requires explicit memory and supervision. OPNet [14] and PermaTrack [19] show that tracking through occlusion benefits from objectives that encourage a persistent latent state. Subsequent work further demonstrates that heavy occlusion demands mechanisms and losses beyond short-term feature aggregation [18, 20].

In autonomous driving perception, several methods improve robustness to *partial* occlusion by strengthening features or reasoning about difficult conditions (e.g., CorrBEV [23] and ReasonNet [15]). However, in end-to-end driving systems, actors that become fully unobservable are typically excluded from supervision and evaluation under standard protocols. Temporal propagation may occasionally recover actors after brief observation gaps, but persistence across longer occlusions is neither explicitly supervised nor evaluated. BeyondSight closes this gap by combining (i) permanence-aware supervision for fully unobservable actors, (ii) architectural state propagation across observability gaps, and (iii) an observability-conditioned evaluation protocol on nuScenes-Permanence.

3 Problem Formulation

Partial Observability. We define the *observability* of an actor using a binary indicator $o_t^i \in \{0, 1\}$ for actor i at time t. Following the nuScenes protocol, observability is approximated by sensor support:

$$o_t^i \approx \mathbb{I}\big(n_{\text{pts},t}^i > 0\big)\,, \tag{1}$$

where $n_{\text{pts},t}^i$ is the number of associated LiDAR and radar returns. A scene exhibits *partial observability* when an actor that was previously observable becomes fully unobservable due to occlusion or sensor resolution.

Object Permanence. We define *object permanence* as the requirement that actors previously observed remain represented even when temporarily unobservable. Formally, a representation satisfies permanence if

$$\forall i, t:\ \Big(o_t^i = 0\ \wedge\ \exists\, t' < t : o_{t'}^i = 1\Big)\ \Rightarrow\ \exists\, r_t^i \in \mathcal{R}_t, \tag{2}$$

where $\mathcal{R}_t$ denotes the set of actor representations maintained at time t and r_t^i corresponds to actor i. Equation 2 expresses permanence as a representation-level constraint that decouples actor existence from instantaneous observation.

In a driving model, $\mathcal{R}_t$ corresponds to the maintained hypothesis set, requiring actor hypotheses to persist across observability gaps. In a dataset, $\mathcal{R}_t$ corresponds to annotated actor states, requiring trajectories to remain defined during unobservable intervals. BeyondSight enforces Eq. 2 architecturally (Sec. 4), while the nuScenes-Permanence benchmark enforces it through supervision and evaluation (Sec. 5).

4 Method

BeyondSight builds upon SparseDrive [17], a sparse query-based end-to-end driving architecture that jointly performs detection, prediction, and planning from multi-view camera inputs. The model maintains a set of instance queries representing dynamic actors and map elements, which are iteratively refined using deformable feature aggregation and temporal attention.

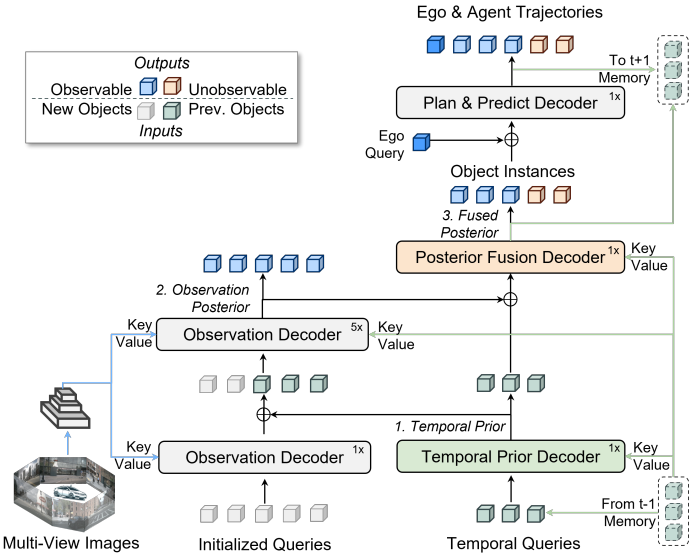

Fig. 2: BeyondSight overview. Actor belief update consists of three stages: (1) a *Temporal Prior Decoder* propagates actor hypotheses from the previous frame, (2) an *Observation Decoder* updates hypotheses using current image features, and (3) a *Posterior Fusion Decoder* merges and refines propagated and observation-conditioned hypotheses.

4.1 BeyondSight Model

To satisfy the actor persistence constraint defined in Sec. 3, BeyondSight introduces a persistence-aware actor inference pipeline while retaining the SparseDrive backbone and objectives. Actor queries are updated through three stages resembling a Bayesian filtering approach:

$$q_t^{\text{prior}} = D_{\text{prior}}(q_{t-1}) \tag{3}$$

$$q_t^{\text{obs}} = D_{\text{obs}}(F_t, q_t^{\text{prior}}) \tag{4}$$

$$q_t^{\text{post}} = D_{\text{fusion}}(q_t^{\text{prior}}, q_t^{\text{obs}}) \tag{5}$$

where F_t denotes BEV features extracted from the perception backbone. The terms prior, observation, and posterior are used in a filtering-inspired sense: they denote learned latent query representations for temporal propagation, image-conditioned update, and fused refinement, rather than calibrated probability distributions.

This formulation separates temporal hypothesis propagation from observation updates, enabling actor representations to persist during observability gaps. Unlike standard sparse-query propagation, where previous detections are refined through the observation decoder and remain tied to recent visual evidence, BeyondSight propagates motion-conditioned hypotheses in an image-feature-free

temporal prior decoder, then fuses them with observation-conditioned queries. This keeps unobservable actors available to prediction and planning without requiring the visual branch to hallucinate non-visible evidence.

Temporal Prior Decoder. The Temporal Prior Decoder propagates actor queries from the previous frame to produce a motion-conditioned hypothesis set. Historical detection queries and motion-predicted queries are fused through a lightweight MLP and refined using transformer-style self-attention operating solely on the queries. This stage updates actor hypotheses based on temporal consistency without accessing image features:

$$q_t^{\text{prior}} = D_{\text{prior}}(q_{t-1}). \tag{6}$$

Observation Decoder. The temporal prior is combined with newly initialized object queries and passed to the standard SparseDrive detection decoder, which attends to BEV features to produce observation-conditioned hypotheses:

$$q_t^{\text{obs}} = D_{\text{obs}}(F_t, q_t^{\text{prior}}). \tag{7}$$

This stage is supervised only using observable ground-truth actors, allowing the propagated prior to maintain hypotheses for actors that remain temporarily unobservable.

Posterior Fusion Decoder. The Posterior Fusion Decoder reconciles propagated and observation-conditioned hypotheses. For each actor, the proposal with higher classification confidence is retained and refined through a final query refinement stage:

$$q_t^{\text{post}} = D_{\text{fusion}}(q_t^{\text{prior}}, q_t^{\text{obs}}). \tag{8}$$

The resulting actor set q_t^{post} forms the scene representation used by the downstream motion prediction and planning modules. Unlike SparseDrive, which supervises only observable actors, BeyondSight also supervises actors that are temporarily unobservable, allowing occluded actors to influence trajectory prediction and planning.

4.2 Optimization

We follow the two-stage training protocol of SparseDrive. In Stage 1, the sparse perception module is trained to learn the scene representation. In Stage 2, perception, motion prediction, and planning modules are trained jointly end-to-end. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}'_{\text{det}} + \mathcal{L}_{\text{map}} + \mathcal{L}'_{\text{motion}} + \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{depth}}. \tag{9}$$

The standard detection loss $\mathcal{L}_{\text{det}}$ supervises the Observation Decoder using observable actors, while the modified detection loss $\mathcal{L}'_{\text{det}}$ supervises the temporal and fusion decoders using both observable and unobservable actors.

Permanence-Aware Supervision. Let G_{obs} and G_{unobs} denote observable and unobservable ground-truth actors at time t, with

$$G_{\text{all}} = G_{\text{obs}} \cup G_{\text{unobs}}. \tag{10}$$

Motion supervision is extended to unobservable actors by applying the motion loss over the full actor set:

$$\mathcal{L}'_{\text{motion}} = \mathcal{L}_{\text{motion}}(G_{\text{all}}). \tag{11}$$

To explicitly decouple existence from observability, the posterior decoder predicts an observability state $\hat{o}_t^i$ for each actor query. This is supervised using binary cross-entropy:

$$\mathcal{L}_{\text{obs}} = \sum_{i \in G_{\text{all}}} \text{BCE}(\hat{o}_t^i, o_t^i). \tag{12}$$

The final detection objective becomes

$$\mathcal{L}'_{\text{det}} = \mathcal{L}_{\text{det}}(G_{\text{all}}) + \lambda_{\text{obs}} \mathcal{L}_{\text{obs}}. \tag{13}$$

5 Permanence Benchmark

Standard perception benchmarks such as nuScenes [1] evaluate only actors that are observable at each timestamp, typically defined by the presence of LiDAR or radar returns. When an actor becomes fully occluded or moves outside sensor range, annotations and evaluation are usually discontinued. As a result, current benchmarks implicitly equate *observability* with *existence*, preventing models from learning or being rewarded for maintaining actor hypotheses during observability gaps. To support training and evaluation of persistence-aware models, we introduce **nuScenes-Permanence**, an extension of the nuScenes dataset that provides supervision for actors that remain present in the scene but are unobservable. The benchmark extension targets approximately continuous actor motion, while abrupt hidden maneuvers that cannot be reliably inferred from surrounding observations are outside its scope.

5.1 Annotation Extension.

We extend actor trajectories through intervals where sensor support is absent by reconstructing missing states offline using physically grounded motion priors. Observable segments remain unchanged, while unobservable intervals are completed using interpolation between observed states or short-horizon extrapolation when actors leave the sensor field of view. Annotated 3D boxes with zero LiDAR or radar points are retained rather than filtered, enabling supervision for heavily occluded actors that remain present in the scene.

Applying this procedure expands the nuScenes annotations from 932k to 1.33M boxes (+30%), transforming the dataset into an occlusion-aware benchmark that supports supervision across observability gaps. The trajectory completion model operates strictly offline and is not used at inference time. BeyondSight must therefore learn to maintain persistent hypotheses without access to completed trajectories.

To validate the generated annotations, we perform a holdout reconstruction study in which observable trajectory intervals are masked, reconstructed with the same offline pipeline, and compared against the original ground truth. We use the resulting reconstruction errors to calibrate an occlusion-duration-dependent matching tolerance for unobservable evaluation. Additional validation details are provided in the supplementary material.

5.2 Observability-Conditioned Evaluation.

Standard nuScenes evaluation penalizes predictions of unobservable actors as false positives, discouraging persistence across occlusion intervals. To evaluate object permanence, we introduce an *observability-conditioned* protocol that partitions ground truth into observable (G_{obs}) and unobservable (G_{unobs}) subsets. Each metric specifies a target ground-truth set G and an ignored set I. Predictions are first matched to G using the standard nuScenes matching procedure; remaining predictions matched to I are removed prior to metric accumulation. This symmetric ignore rule prevents cross-penalization between observable and unobservable actors. In addition to the standard detection metric mAP, we report:

- mAP_{obs}: performance on continuously observable actors,
- $\text{mAP}_{\text{unobs}}$: performance during unobservable intervals,
- mAP_{all}: performance across the full actor set.

Because reference trajectories for unobservable actors are partially extrapolated, we use an adaptive matching tolerance that accounts for accumulated motion uncertainty during unobservability. This tolerance reduces to the standard nuScenes threshold when the unobservability duration approaches zero, ensuring compatibility with the official evaluation protocol. Details of the trajectory completion model and tolerance estimation are provided in the supplementary material.

6 Experiments

6.1 Experimental Setup

Dataset. We evaluate on nuScenes [1], a large-scale autonomous driving dataset containing 1,000 driving scenes of approximately 20 s each. Following standard practice, we use the official train/validation split (700/150 scenes). Each keyframe provides synchronized multi-view imagery from six cameras together with 3D

bounding box annotations at 2 Hz for ten object classes. In addition to the standard dataset, we evaluate on **nuScenes-Permanence**, our extended benchmark introduced in Section 5. The extension augments nuScenes with supervision for actors that remain physically present but become temporarily unobservable due to occlusion, sensor range limits, or field-of-view constraints. This extension increases the number of annotated actor states by approximately 30%, enabling systematic evaluation of persistence-aware models.

Metrics. We report the standard nuScenes metrics to ensure comparability with prior work. We evaluate *detection* using mAP and NDS together with the standard nuScenes error metrics mATE, mASE, mAOE, mAVE, and mAAE [1]. *Tracking* performance is evaluated using AMOTA, AMOTP, recall, and ID switches. For *motion forecasting* we report minADE, minFDE, Miss Rate (MR), and End-to-end Prediction Accuracy (EPA) following the VIP3D protocol [3] and UniAD-style evaluation [5]. *Planning* quality is measured using open-loop trajectory error and collision rate following the VAD evaluation protocol [8].

To evaluate persistence under partial observability, we additionally report the proposed *observability-conditioned metrics* on nuScenes-Permanence: mAP_{obs}, $\text{mAP}_{\text{unobs}}$, mAP_{all}. These metrics isolate performance on observable and unobservable actors while applying symmetric ignore rules to prevent penalization. For unobservable intervals whose reference trajectories are extrapolated, we use the adaptive matching tolerance described in Section 5.

Implementation Details. BeyondSight is implemented on top of SparseDrive and retains its backbone architecture, sparse scene representation, motion prediction, and planning modules. Unless otherwise specified, all architectural hyperparameters follow the corresponding SparseDrive configuration. The Temporal Prior Decoder and Posterior Fusion Decoder are implemented as a lightweight transformer decoder using self-attention, two-layer MLPs, and a detection head operating on the query embeddings, introducing less than 5% additional parameters.

Training follows a two-stage protocol. First, the sparse perception module is trained to learn the scene representation. Second, perception, motion prediction, and planning modules are trained jointly in an end-to-end manner. We extend detection and motion supervision to unobservable timestamps and introduce an observability classification loss with weight $\lambda_{\text{obs}} = 1.0$. Models are trained using AdamW with weight decay 10^{-3}, cosine learning-rate scheduling, and 500 warm-up iterations. Stage 1 uses learning rate 4×10^{-4}, while Stage 2 uses learning rate 3×10^{-4}. The image backbone learning-rate multiplier is set to 0.5 in Stage 1 and 0.1 in Stage 2. Full optimization details are provided in the supplementary material.

6.2 Main Result

We compare BeyondSight to state-of-the-art vision-based end-to-end driving baselines that jointly model perception, forecasting, and planning. Our primary

baseline is SparseDrive [17], since BeyondSight directly extends its sparse-query representation and training protocol. We additionally compare to representative end-to-end driving systems including UniAD [5], VAD [8], and recent efficient E2E-AD approaches when supported by public code and evaluation settings. All methods are evaluated under the same nuScenes splits and metrics; where necessary, we re-train models under a unified pipeline to ensure fair comparison.

End-to-end Performance. We first report standard nuScenes leaderboard metrics to ensure comparability with prior work (Tables 1 and 2). BeyondSight consistently improves the strongest SparseDrive baseline across multiple tasks while maintaining compatibility with the official evaluation protocol.

Table 1: Open-loop planning performance on the nuScenes validation set. Lower L2 error and collision rate are better.

Method	$L2_{1s}\downarrow$	$L2_{2s}\downarrow$	$L2_{3s}\downarrow$	$L2_{avg}\downarrow$	CR_{1s} (%) $\downarrow$	CR_{2s} (%) $\downarrow$	CR_{3s} (%) $\downarrow$	CR_{avg} (%) $\downarrow$
ST-P3	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Base	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
SparseDrive	0.29	0.58	0.96	0.61	0.01	0.05	0.18	0.08
BridgeAD	0.28	0.55	0.92	0.58	0.00	0.04	0.20	0.08
MomAD	0.31	0.57	0.91	0.60	0.01	0.05	0.22	0.09
BeyondSight	**0.26**	**0.51**	**0.85**	**0.54**	**0.01**	**0.04**	**0.16**	**0.07**

Table 2: Perception, tracking, and prediction performance on the nuScenes validation set. Best results are in bold.

Method	Backbone	mAP ↑	NDS ↑	AMOTA ↑	minADE ↓	EPA ↑
SparseDrive	ResNet50	0.415	0.526	0.372	0.610	0.492
BridgeAD	ResNet50	0.423	0.534	0.398	0.620	0.500
MomAD	ResNet50	0.423	0.531	0.391	0.610	0.499
BeyondSight	ResNet50	**0.427**	**0.536**	**0.401**	**0.610**	**0.502**

On perception, BeyondSight improves detection from 0.415 to 0.427 mAP and from 0.526 to 0.536 NDS. Tracking performance also improves, increasing AMOTA from 0.372 to 0.401. Motion forecasting performance remains competitive while slightly improving EPA. These results indicate that persistence-aware reasoning improves the quality and stability of the scene representation without degrading standard perception metrics.

BeyondSight also improves downstream planning performance. Planning error decreases from 0.61 to 0.54 $L2_{\text{avg}}$, while the collision rate decreases from

0.08% to 0.07%. Importantly, these gains are achieved without modifying the planner itself. Instead, improvements arise from the richer scene representation produced by permanence-aware actor modeling.

Crucially, the permanence-aware model does not incur additional false positives under the official nuScenes evaluation protocol. Standard mAP and NDS remain competitive with state-of-the-art baselines, demonstrating that maintaining persistent actor hypotheses does not negatively impact conventional benchmark performance.

Object Permanence Performance. We next evaluate BeyondSight using the proposed observability-conditioned protocol on the nuScenes-Permanence benchmark. Table 3 reports detection performance for observable actors, unobservable actors, and the full actor set.

Table 3: Observability-conditioned detection and prediction performance on the nuScenes-Permanence validation set.

Method	$\text{mAP}_{\text{all}} \uparrow$	$\text{NDS}_{\text{all}} \uparrow$	$\text{minADE}_{\text{all}} \downarrow$	$\text{EPA}_{\text{all}} \uparrow$
SparseDrive	0.389	0.514	0.642	0.457
BeyondSight	**0.413**	**0.519**	**0.582**	**0.481**
Method	$\text{mAP}_{\text{obs}} \uparrow$	$\text{NDS}_{\text{obs}} \uparrow$	$\text{minADE}_{\text{obs}} \downarrow$	$\text{EPA}_{\text{obs}} \uparrow$
SparseDrive	0.415	0.526	**0.610**	0.492
BeyondSight	**0.421**	**0.528**	0.625	**0.496**
Method	$\text{mAP}_{\text{unobs}} \uparrow$	$\text{NDS}_{\text{unobs}} \uparrow$	$\text{minADE}_{\text{unobs}} \downarrow$	$\text{EPA}_{\text{unobs}} \uparrow$
SparseDrive	0	0.255	0.615	0
BeyondSight	**0.249**	**0.306**	**0.479**	**0.285**

BeyondSight substantially improves performance during occlusion. The model achieves the highest $\text{mAP}_{\text{unobs}}$, improving over the strongest baseline by a large margin. In contrast, performance on continuously observable actors remains nearly unchanged. This demonstrates that permanence-aware supervision specifically improves reasoning about unobservable actors without degrading standard detection performance. The aggregated metric mAP_{all} further confirms improved holistic scene understanding. By maintaining persistent hypotheses across observability gaps, BeyondSight produces a more complete representation of the dynamic environment.

6.3 Ablation Study

We conduct ablation experiments to analyze the contribution of the key components introduced in BeyondSight. All ablations are evaluated on the nuScenes

validation split using the same training protocol as the full model. Table 4 reports cumulative component additions under two annotation settings: standard nuScenes annotations and nuScenes-Permanence annotations.

Table 4: Ablation study of BeyondSight components under standard and permanence-aware annotations.

Configuration	mAP_{obs} ↑	mAP_{unobs} ↑	mAP_{all} ↑	$L2_{avg}$ ↓	CR_{avg} ↓
Trained on standard nuScenes annotations					
SparseDrive	0.415	0.000	0.389	0.61	0.08
+ Temporal prior	0.417	0.000	0.390	0.60	0.08
+ Posterior fusion	0.413	0.000	0.388	0.61	0.08
Trained on nuScenes-Permanence annotations					
SparseDrive	0.403	0.021	0.388	0.61	0.08
+ Temporal prior	0.406	0.126	0.399	0.58	0.08
+ Posterior fusion	0.415	0.194	0.405	0.56	**0.07**
+ Unobs. supervision	0.418	0.213	0.410	0.55	**0.07**
BeyondSight	**0.421**	**0.249**	**0.413**	**0.54**	**0.07**

Under standard nuScenes annotations, unobservable actors are not explicitly supervised, and all configurations obtain zero mAP_{unobs}. Adding the Temporal Prior Decoder slightly improves mAP_{obs}, mAP_{all}, and planning error, but does not enable detection of fully unobservable actors by itself. With nuScenes-Permanence annotations, the same temporal prior provides a substantial gain on mAP_{unobs}, showing that temporal propagation is most effective when paired with permanence-aware supervision.

The Posterior Fusion Decoder further improves both observable and unobservable detection by reconciling propagated hypotheses with observation-conditioned queries. Adding unobservable-actor supervision improves the persistence of occluded actors and reduces planning error. The full BeyondSight model achieves the best overall performance, improving mAP_{unobs} from 0.021 to 0.249 over the SparseDrive baseline trained on nuScenes-Permanence, while also improving mAP_{obs}, mAP_{all}, $L2_{avg}$, and CR_{avg}.

6.4 Planning under Occlusion

We qualitatively evaluate BeyondSight in scenarios involving partial observability and prolonged occlusions. Figure 3 shows a planning-relevant case where an actor becomes fully occluded by another vehicle. SparseDrive drops the hidden actor once direct evidence disappears, causing the ego plan to intersect the actor's ground-truth future trajectory. BeyondSight maintains a persistent hypothesis for the occluded actor and produces a plan with additional clearance. This example illustrates how permanence-aware scene representations can pro-

vide useful context to downstream planning when hidden actors remain decision-relevant.

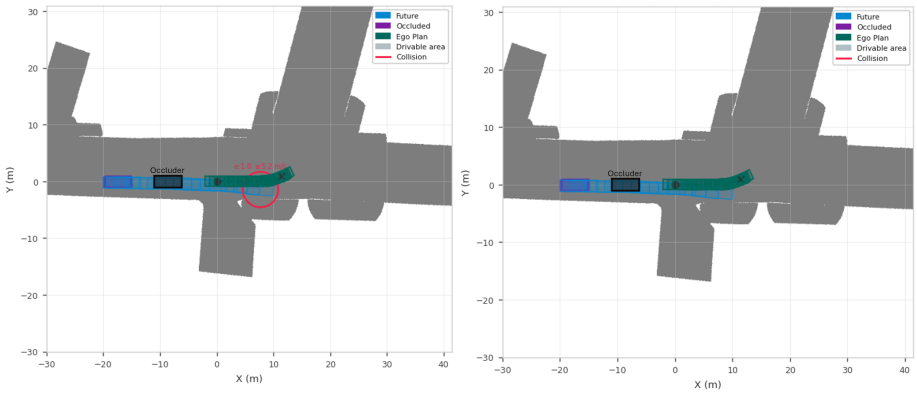

Fig. 3: Planning under full occlusion. Left: SparseDrive drops the occluded actor once visual evidence disappears, causing the ego plan to intersect the actor's ground-truth future trajectory. Right: BeyondSight maintains a persistent hypothesis for the hidden actor and produces a safer plan. The occluder, hidden actor, actor future, and ego plan are highlighted for clarity.

To quantify this behavior beyond a single qualitative example, we evaluate a high-occlusion subset containing 600 validation samples with occluded dynamic agents whose future trajectories approach the ego plan within 12 m over a 4 s horizon. On this subset, SparseDrive degrades to 0.64 $L2_{avg}$ and 0.163 CR_{avg}, while BeyondSight achieves 0.56 $L2_{avg}$ and 0.08 CR_{avg}. These results suggest that persistent actor hypotheses are most useful when unobservable actors remain relevant to the ego plan.

7 Conclusion

We presented BeyondSight, a permanence-aware end-to-end driving framework designed to operate reliably under partial observability. Unlike existing systems that implicitly couple actor existence to instantaneous observations, BeyondSight maintains persistent actor hypotheses across occlusion intervals through temporally propagated and observation-conditioned updates. We further introduced nuScenes-Permanence, an extension of the nuScenes benchmark that enables supervision and evaluation of actors during unobservable intervals through observability-conditioned metrics. This benchmark provides the first systematic evaluation protocol for object permanence in end-to-end autonomous driving.

Limitations. BeyondSight's persistent hypotheses can become stale during long occlusions, leading to false persistence or localization drift when actors

deviate from the learned temporal prior. nuScenes-Permanence unobservable annotations target occlusions with approximately continuous motion and may not capture abrupt hidden maneuvers such as sudden stops, turns, or intent changes. Future work should explore uncertainty-aware memory, simulation-based unobservable annotations with privileged labels, and multi-observer datasets for direct occlusion evaluation.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
2. Cheong, B., Wang, L., Papais, S., Waslander, S.L.: Scatr: Mitigating new instance suppression in lidar-based tracking-by-attention via second chance assignment and track query dropout. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3330–3339 (2026)
3. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5496–5506 (2023)
4. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: European Conference on Computer Vision. pp. 533–549. Springer (2022)
5. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17853–17862 (2023)
6. Jia, X., Gao, Y., Chen, L., Yan, J., Liu, P.L., Li, H.: DriveAdapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
7. Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. arXiv preprint arXiv:2503.07656 (2025)
8. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)
9. Li, P., Cui, D.: Navigation-guided sparse scene representation for end-to-end autonomous driving. arXiv preprint arXiv:2409.18341 (2024)
10. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
11. Papais, S., Ren, R., Waslander, S.: Swtrack: Multiple hypothesis sliding window 3d multi-object tracking. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 4939–4945. IEEE (2024)
12. Papais, S., Wang, L., Cheong, B., Waslander, S.L.: Foresight: Multi-view streaming joint object detection and trajectory forecasting. In: Proceedings of the IEEE/CVF

International Conference on Computer Vision (ICCV). pp. 25474–25484 (October 2025)
13. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision (ECCV). pp. 194–210 (2020). https://doi.org/10.1007/978-3-030-58568-6_12
14. Shamsian, A., Kleinfeld, O., Globerson, A., Chechik, G.: Learning object permanence from video. In: European Conference on Computer Vision. pp. 35–50. Springer (2020)
15. Shao, H., Wang, L., Chen, R., Waslander, S.L., Li, H., Liu, Y.: Reasonnet: End-to-end driving with temporal and global reasoning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13723–13733 (2023)
16. Song, Z., Jia, C., Liu, L., Pan, H., Zhang, Y., Wang, J., Zhang, X., Xu, S., Yang, L., Luo, Y.: Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22432–22441 (June 2025)
17. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8795–8801. IEEE (2025)
18. Tokmakov, P., Jabri, A., Li, J., Gaidon, A.: Object permanence emerges in a random walk along memory. In: International Conference on Machine Learning. pp. 21506–21519. PMLR (2022)
19. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10860–10869 (2021)
20. Van Hoorick, B., Tokmakov, P., Stent, S., Li, J., Vondrick, C.: Tracking through containers and occluders in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13802–13812 (2023)
21. Wang, L., Lavoie, M.A., Papais, S., Nisar, B., Chen, Y., Ding, W., Ivanovic, B., Shao, H., Abuduweili, A., Cook, E., et al.: Trends in motion prediction toward deployable and generalizable autonomy: A revisit and perspectives. Foundations and Trends® in Robotics **13**(1-2), 1–269 (2026)
22. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3621–3631 (2023)
23. Xue, Z., Guo, M., Fan, H., Zhang, S., Zhang, Z.: Corrbev: Multi-view 3d object detection by correlation learning with multi-modal prototypes. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27413–27423 (2025)
24. Zhang, B., Song, N., Jin, X., Zhang, L.: Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6854–6863 (June 2025)