LATENTSIGHTDRIVE: PROGRESSIVE FORESIGHT INTERNALIZATION FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Anticipating dynamic interactions is crucial for safe autonomous driving. While future reasoning provides valuable foresight, not all future evidence is equally relevant to the current ego-trajectory decision. The challenge is to identify decision-relevant future evidence and internalize it into the planner, preserving its benefits without the future-reasoning process at inference. To address this challenge, we propose LatentSightDrive (LSDrive), a *Select-and-Internalize* paradigm that progressively transfers selected foresight into the planner's latent space. LSDrive projects external future evidence and planner representations into a common foresight-slot space, where Hierarchical Foresight Selection determines when and what to internalize. Through weighted alignment, this selective process progressively shapes the planner's latent space, advancing from uniform bootstrapping to selective refinement. At inference, only the direct observation-to-trajectory planner is retained. Experiments on NAVSIM and nuScenes show consistent improvements over uniform transfer across latent-world and visual-CoT settings. On nuScenes, LSDrive outperforms explicit visual-CoT planning, lowering L2 error by 9.3% and collision rate by 31.7%, while reducing inference time by 31.7%.

1 INTRODUCTION

End-to-end autonomous driving aims to predict future ego trajectories directly from sensor observations, ego-state information, and navigation commands (Codevilla et al., 2018; Chitta et al., 2023; Hu et al., 2022; Shao et al., 2023; Hu et al., 2023; Jiang et al., 2023; Chen et al., 2024a;c). Recent advances further enrich this observation-to-trajectory mapping with pretrained semantic knowledge and reasoning capabilities, particularly through VLM- and VLA-based planners (Sima et al., 2023; Hwang et al., 2024; Jiang et al., 2024; Tian et al., 2025; Zhou et al., 2025). However, reliable planning requires more than understanding the current scene: the planner must also anticipate how surrounding agents and the environment may evolve before selecting an ego trajectory. Motivated by this need, recent world-model-based planners incorporate future evidence in various forms, including explicit future scenes (Wang et al., 2024; Li et al., 2026b), visual chain-of-thought (CoT) traces (Zeng et al., 2025), and latent world representations (Min et al., 2024; Li et al., 2025a;b; Zheng et al., 2025; Li et al., 2026a).

Among planners that use future evidence at inference, two representative designs are *explicit reasoning rollouts* and *latent world representations*. For example, FutureSightDrive (FSDrive) autoregressively generates a future frame that serves as a visual spatio-temporal CoT before predicting the ego trajectory (Zeng et al., 2025). Such explicit rollouts provide an intuitive representation of scene evolution, but require additional generation before planning, introducing extra computation and inference latency. Alternatively, SGDrive encodes hierarchical driving knowledge, including anticipated scene evolution and short-term goals, into continuous world queries that condition its diffusion planner (Li et al., 2026a). Although latent representations provide a more compact interface between future reasoning and planning, both representative designs retain an intermediate future-aware representation as a planning input at inference.

We revisit this design choice. Although future reasoning provides valuable cues for planning, it need not necessarily remain an explicit inference-time dependency (Li et al., 2026b; Lin et al., 2026). An alternative is to use an external future-reasoning model as a training-time source of foresight, internalize its useful knowledge into the planner, and retain only the direct observation-to-trajectory pathway at deployment. However, simply transferring all available future information is also undesirable, because the utility of foresight is inherently decision-dependent. Routine scenes may require little external guidance, unreliable predictions should not dominate learning, and only a subset of future components may influence the current ego trajectory (Feng et al., 2024; Yang et al., 2026; Lin et al., 2026). Therefore, the key challenge is not merely to predict more of the future or to uniformly transfer it, but to determine *when* foresight should guide the planner and *what* future information should be internalized.

This leads to the central question of this work: *can a planner benefit from future reasoning during training without depending on it at inference, while internalizing the foresight that matters for the current driving decision?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

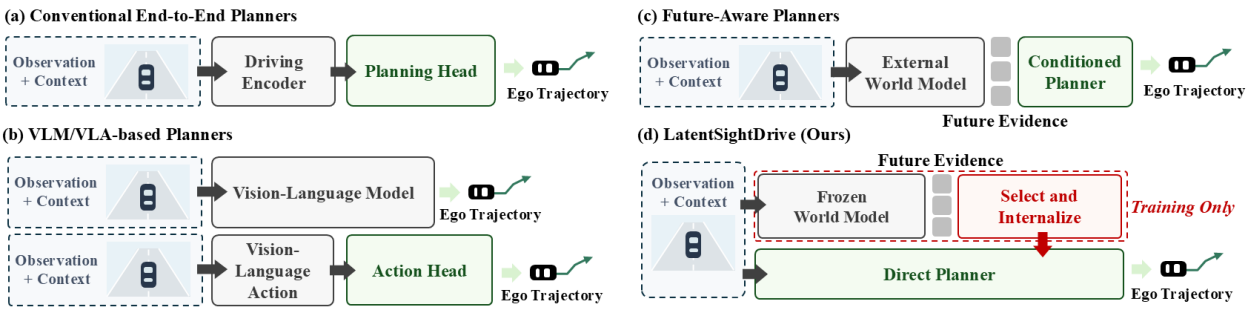

Figure 1: **Paradigm comparison between prior methods and LatentSightDrive.** Direct planner predicts ego trajectories from current observations and driving context, using internalized foresight without requiring an external world model or explicit future reasoning at inference.

To address this challenge, we propose **LatentSightDrive (LSDrive)**, a framework for **Progressive Foresight Internalization**. As illustrated in Fig. 1, LSDrive follows a *Select-and-Internalize* paradigm: it uses external future reasoning only during training, selectively internalizes planning-relevant foresight into the planner, and predicts ego trajectories without the external reasoning pathway at inference. Specifically, LSDrive performs **Hierarchical Foresight Selection** to determine *when* external foresight should guide learning and *which* foresight components are most relevant to the planning decision. The internalization proceeds progressively: an initial bootstrapping stage aligns external and internal foresight representations, followed by selective refinement that concentrates supervision on difficult scenes and planning-relevant foresight components. After training, the external future-reasoning model and all guidance-specific modules are removed, leaving only the observation-to-trajectory planning path for deployment.

Our contributions are summarized as follows:

- We introduce **LatentSightDrive (LSDrive)** and a *Select-and-Internalize* paradigm that reformulates future reasoning from an inference-time dependency into a training-time source of foresight. This enables a direct planner to acquire future-aware knowledge without retaining the external reasoning pathway at deployment.
- We propose **Hierarchical Foresight Selection** for foresight internalization at the scene and slot levels. **Scene-Adaptive Guidance** determines *when* external foresight should guide learning, while **Planning-Relevant Foresight Scoring** determines *which* foresight components are most relevant to the ego-trajectory decision.
- We instantiate LSDrive with explicit visual-CoT reasoning and latent world representations across planning architectures and datasets. Experiments show that selective internalization consistently improves over uniform foresight transfer, while enabling the external future-reasoning pathway to be removed at inference.

2 Related Work

2.1 Future-Aware Autonomous Driving

Future-aware autonomous driving incorporates predictive knowledge of scene evolution into representation learning and trajectory planning. DriveWorld learns spatio-temporal driving representations through world-model-based pretraining (Min et al., 2024), while LAW employs action-conditioned future feature prediction as auxiliary supervision for end-to-end planning (Li et al., 2025a). Other methods explicitly introduce predicted future information into the planning process: DriveWM generates action-conditioned multiview futures for maneuver evaluation (Wang et al., 2024), whereas WoTE and World4Drive use predicted future BEV or latent states for trajectory evaluation and selection (Li et al., 2025b; Zheng et al., 2025). Future evidence can also serve as an intermediate planning representation. FSDrive generates a future frame as visual spatio-temporal CoT and conditions trajectory prediction on the imagined scene (Zeng et al., 2025), while SGDrive organizes future driving knowledge into hierarchical world queries that condition a diffusion planner (Li et al., 2026a). These approaches demonstrate complementary ways of exploiting future prediction for autonomous driving. These representations differ substantially in granularity, semantics, and computational form. Rather than introducing another future-prediction mechanism, LSDrive focuses on how externally generated future evidence can be selectively internalized into a direct planner according to its relevance to the current planning decision.

2.2 Reasoning Internalization and Knowledge Transfer

Reasoning internalization seeks to absorb explicit intermediate reasoning into a model's learned representations or computation, reducing the need to reproduce the same reasoning process at inference (Deng et al., 2023; Shen et al.,

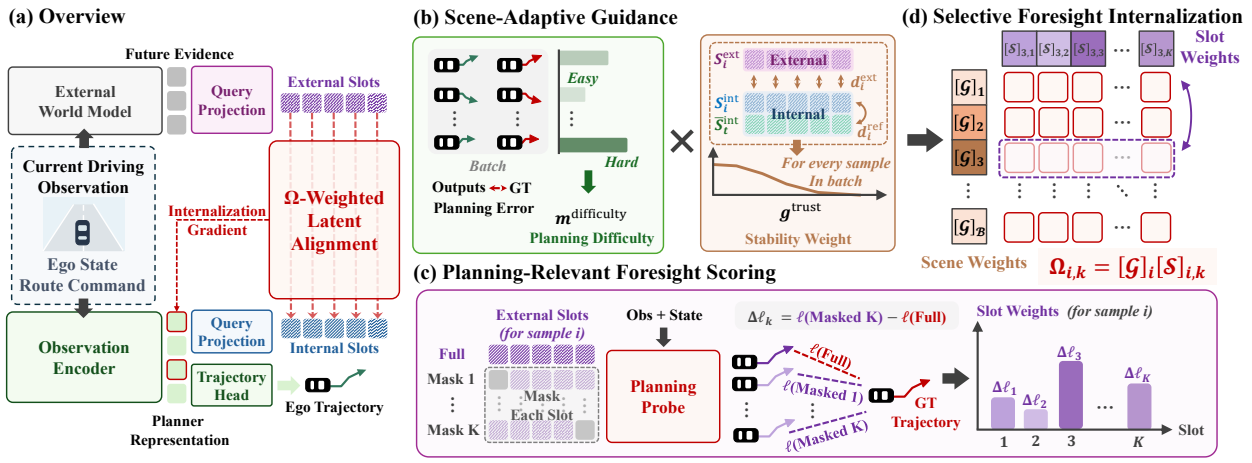

Figure 2: **Overview of LatentSightDrive (LSDrive).** During training, LSDrive constructs external and internal foresight representations and uses hierarchical guidance to derive selection weights $\mathbf{\Omega}$. Through $\mathbf{\Omega}$-weighted latent alignment, LSDrive selectively internalizes decision-relevant foresight into the planner. At inference, only the planner is retained, eliminating the need for full future reasoning.

2025). For example, Implicit CoT distills explicit chain-of-thought reasoning into internal hidden states, enabling prediction without generating the corresponding reasoning sequence (Deng et al., 2023). Related knowledge-transfer paradigms have also been explored in autonomous driving. PlanKD transfers planning-relevant knowledge from a larger planner to a compact student through information-bottleneck-based distillation (Feng et al., 2024). More closely related to our setting, WPT transfers world-model-guided planning knowledge into a direct policy through planning-query alignment and world reward distillation (Jiang et al., 2026). LSDrive differs in both the transferred representation and the mechanism used to allocate supervision: it operates on external future-evidence slots and explicitly models their decision-dependent utility through scene-level guidance and slot-level planning relevance. This enables progressive, selective internalization across both explicit visual reasoning and latent world representations.

3 Method

3.1 Overview and Problem Formulation

Given the current driving observation and context $\mathcal{O}_i$, a standard end-to-end planner predicts the future ego trajectory as $\widehat{\mathbf{Y}}_i = f_\theta^{\text{planner}}(\mathcal{O}_i)$. In contrast, world-model-enhanced planners obtain foresight from an external future-reasoning model, formulated as $\mathbf{E}_i = \mathcal{W}^{\text{external}}(\mathcal{O}_i)$, and condition trajectory prediction on this evidence as $\widehat{\mathbf{Y}}_i = f_\theta^{\text{planner}}(\mathcal{O}_i, \mathbf{E}_i)$. $\mathbf{E}_i$ may correspond to explicit future scenes or visual reasoning traces (Wang et al., 2024; Zeng et al., 2025), or latent world representations (Li et al., 2025a; Zheng et al., 2025; Li et al., 2026a).

LSDrive instead treats $\mathbf{E}_i$ as a *training-time source of foresight* and selectively internalizes useful future knowledge into the planner. Its learning paradigm is summarized as

$$\underbrace{\mathcal{O}_i \longrightarrow \mathcal{W}^{\text{external}} \longrightarrow \mathbf{E}_i \xrightarrow[\mathcal{S}\text{: } \textit{what} \text{ to internalize}]{\mathcal{G}\text{: } \textit{when} \text{ to guide}} f_\theta^{\text{planner}}}_{\text{Select-and-Internalize Training}} \quad \Longrightarrow \quad \underbrace{\mathcal{O}_i \longrightarrow f_{\theta^*}^{\text{planner}} \longrightarrow \widehat{\mathbf{Y}}_i}_{\text{Direct Inference}}. \tag{1}$$

Here, $\mathbf{E}_i$ provides training-time supervision, rather than serving as an additional input to its trajectory prediction path. After training, the external future-reasoning model and guidance-specific modules are removed.

LSDrive realizes this paradigm through **Progressive Foresight Internalization**. It first bootstraps uniform alignment between external and planner-side foresight representations and then selectively refines the internalization according to scene-level guidance and slot-level planning relevance. Scene-Adaptive Guidance $\mathcal{G}$ determines *when* foresight should guide learning, while Planning-Relevant Foresight Scoring $\mathcal{S}$ determines *which* foresight components should be emphasized. Together, they define selection weights as:

$$\Omega_{i,k} = [\mathcal{G}]_i [\mathcal{S}]_{i,k}, \qquad \mathbf{\Omega} \in [0,1]^{|\mathcal{B}| \times K}, \tag{2}$$

where $\mathcal{B}$ denotes the current mini-batch, K is the number of foresight slots, $[\mathcal{G}]_i$ controls scene-level guidance, and $[\mathcal{S}]_{i,k}$ distributes this guidance according to slot-level planning relevance.

Via $\mathbf{\Omega}$-weighted latent alignment between external and planner-side foresight representations, LSDrive internalizes planning-relevant foresight, effectively shaping the planner's latent representation during training. The overview of LSDrive is illustrated in Fig. 2.

3.2 Progressive Foresight Internalization

LSDrive progressively internalizes external foresight in two stages. During *Bootstrapping*, Uniform Internalization establishes an initial alignment between external and planner-side foresight representations. During *Refinement*, Hierarchical Foresight Selection allocates guidance according to scene-level planning difficulty and slot-level planning relevance. Both stages operate on a common foresight-slot space, which provides the representation interface for alignment and selection. We first introduce Foresight Slot Projection, followed by hierarchical selection and the internalization objective.

3.2.1 Foresight Slot Projection

To enable foresight internalization across heterogeneous future representations, we construct compact slot-based representations on both the planner and external reasoning sides. The planner-side slots encode future-relevant information inferred from the current observation, while the external slots provide training-time foresight guidance.

For sample i, let

$$\mathbf{H}_i^{\text{planner}} \in \mathbb{R}^{N_i \times d_{\text{planner}}}, \qquad \mathbf{E}_i \in \mathbb{R}^{L_i \times d_{\text{external}}}, \tag{3}$$

denote the planner representation and external foresight representation, respectively. We construct the planner-side *internal* foresight slots as

$$\mathbf{S}_i^{\text{int}} = \Pi_{\psi_{\text{int}}}\left(\mathbf{H}_i^{\text{planner}}; \mathbf{Q}^{\text{future}}\right), \qquad \mathbf{S}_i^{\text{int}} \in \mathbb{R}^{K \times d_{\text{foresight}}}, \tag{4}$$

where $\mathbf{Q}^{\text{future}} = \{\mathbf{q}_k\}_{k=1}^{K}$ is a set of learnable foresight queries. Through cross-attention over $\mathbf{H}_i^{\text{planner}}$, these queries extract K latent slots encoding future-relevant information from the current driving context.

The *external* foresight is compacted into the same slot dimensionality:

$$\mathbf{S}_i^{\text{ext}} = \Pi_{\psi_{\text{ext}}}\left(\mathbf{E}_i\right), \qquad \mathbf{S}_i^{\text{ext}} \in \mathbb{R}^{K \times d_{\text{foresight}}}, \tag{5}$$

where $\Pi_{\psi_{\text{ext}}}$ denotes a lightweight feature adaptation and sequence compaction module.

Slot-wise semantic correspondence is not assumed a priori. Instead, *Bootstrapping* encourages internal and external slots to become aligned before selective refinement is activated. The resulting representations provide a common interface for subsequent Hierarchical Foresight Selection. Additional projection details are provided in Appendix A.1.

3.2.2 Hierarchical Foresight Selection

Hierarchical Foresight Selection decomposes foresight selection into two complementary levels. **Scene-Adaptive Guidance** determines *when* external foresight should guide learning, while **Planning-Relevant Foresight Scoring** determines *which* foresight components should receive greater emphasis.

Scene-Adaptive Guidance External foresight is not equally beneficial across driving scenes. Scene-Adaptive Guidance therefore prioritizes samples that remain difficult for the current planner while applying a latent space stability constraint to suppress external foresight that is excessively inconsistent with the planner's evolving representation.

For sample i, let $\ell_i^{\text{planning}} = \mathcal{L}_{\text{traj}}(\widehat{\mathbf{Y}}_i, \mathbf{Y}_i)$ denote its trajectory prediction loss. We identify difficult samples using the q-th batch quantile:

$$m_i^{\text{difficulty}} = \mathbb{I}\left[\ell_i^{\text{planning}} \geq \text{Quantile}_q\left(\{\ell_j^{\text{planning}}\}_{j \in \mathcal{B}}\right)\right], \tag{6}$$

where $\mathcal{B}$ denotes the current mini-batch and $q \in (0, 1)$ sets the quantile threshold. $q = 0.7$ selects the hardest 30% of samples in the mini-batch.

Planning difficulty identifies under-optimized ego-trajectory decisions, but a difficult scene may also exhibit a large discrepancy between external and internal foresight, making strong alignment potentially disruptive to ongoing planning optimization. To stabilize internalization, we introduce a soft, scene-adaptive weighting inspired by the trust-region principle (Schulman et al., 2015). Just as trust-region limits policy updates to avoid catastrophic deviations, our mechanism dynamically constrains alignment strength to prevent latent space disruption. Specifically, it evaluates the

external-internal discrepancy relative to a running planner-side reference, assigning lower weights to samples with larger mismatches. This soft gating allows the planner to internalize reliable future guidance while safely attenuating the influence of highly divergent external representations.

To achieve this mechanism, we maintain a detached running reference $\overline{\mathbf{S}}_t^{\text{int}}$ of the evolving planner-side foresight representation and compare external-internal discrepancy against the planner's endogenous variation. The running-reference update is detailed in Appendix A.2. Specifically, we measure

$$d_i^{\text{ext}} = \frac{1}{K}\sum_{k=1}^{K}\left[1-\cos\left(\mathbf{s}_{i,k}^{\text{int}},\mathbf{s}_{i,k}^{\text{ext}}\right)\right], \quad d_i^{\text{ref}} = \frac{1}{K}\sum_{k=1}^{K}\left[1-\cos\left(\mathbf{s}_{i,k}^{\text{int}},\bar{\mathbf{s}}_{t,k}^{\text{int}}\right)\right]. \tag{7}$$

Here, $\cos(\cdot,\cdot)$ denotes cosine similarity. d_i^{ext} measures the discrepancy between planner-side and external foresight, while d_i^{ref} provides an adaptive scale for the planner's own representation variation. Their relative magnitude defines a soft stability weight:

$$g_i^{\text{trust}} = \sigma\left(\frac{d_i^{\text{ref}} - d_i^{\text{ext}}}{\tau_{\text{trust}}}\right), \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\tau_{\text{trust}} = 0.1$ controls transition smoothness.

The resulting scene-level guidance weight is

$$[\mathcal{G}]_i = m_i^{\text{difficulty}}\, g_i^{\text{trust}}, \tag{9}$$

where planning difficulty $m_i^{\text{difficulty}}$ determines guidance priority, while the stability weight g_i^{trust} moderates its influence on representation learning.

Planning-Relevant Foresight Scoring Scene-Adaptive Guidance determines whether a scene should receive external guidance, but different foresight components within the same scene may contribute unequally to the ego decision. We therefore estimate slot-level planning relevance using a lightweight, training-only trajectory probe. The probe provides a unified trajectory-based assessment of planning relevance across different external future-reasoning models, without relying on their model-specific planning interfaces. Details of trajectory probe are provided in Appendix A.3.

Given the external foresight slots $\mathbf{S}_i^{\text{ext}}$ and a compact driving condition $\mathbf{c}_i^{\text{driving}}$ containing the ego state and route command, the probe predicts

$$\widehat{\mathbf{Y}}_i^{\text{full}} = q_\phi^{\text{probe}}\left(\mathbf{S}_i^{\text{ext}}, \mathbf{c}_i^{\text{driving}}\right). \tag{10}$$

The probe is optimized using auxiliary trajectory supervision:

$$\mathcal{L}_{\text{probe}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\mathcal{L}_{\text{traj}}\left(\widehat{\mathbf{Y}}_i^{\text{full}}, \mathbf{Y}_i\right). \tag{11}$$

To estimate the contribution of slot k, we perform a leave-one-slot-out intervention by zero-masking $\mathbf{s}_{i,k}^{\text{ext}}$ while keeping the remaining slots unchanged. Denoting the resulting representation by $\mathbf{S}_{i,\setminus k}^{\text{ext}}$, the same probe predicts

$$\widehat{\mathbf{Y}}_{i,\setminus k} = q_\phi^{\text{probe}}\left(\mathbf{S}_{i,\setminus k}^{\text{ext}}, \mathbf{c}_i^{\text{driving}}\right). \tag{12}$$

We define the planning relevance of slot k as the increase in trajectory loss caused by removing it:

$$u_{i,k} = \left[\mathcal{L}_{\text{traj}}\left(\widehat{\mathbf{Y}}_{i,\setminus k}, \mathbf{Y}_i\right) - \mathcal{L}_{\text{traj}}\left(\widehat{\mathbf{Y}}_i^{\text{full}}, \mathbf{Y}_i\right)\right]_+, \tag{13}$$

where $[\cdot]_+ = \max(0,\cdot)$. A larger value indicates that trajectory prediction is more sensitive to the corresponding foresight slot.

The slot-level selection weight is then

$$[\mathcal{S}]_{i,k} = \frac{u_{i,k}+\epsilon}{\sum_{j=1}^{K} u_{i,j} + K\epsilon}, \tag{14}$$

where $\sum_{k=1}^{K}[\mathcal{S}]_{i,k} = 1$. When no component exhibits a clear contribution, the distribution approaches uniform weighting.

3.2.3 SELECTIVE FORESIGHT INTERNALIZATION

Using the hierarchical guidance weights $\Omega_{i,k}$ in Eq. 2, we internalize external foresight through a weighted latent-alignment objective:

$$\mathcal{L}_{\text{internalization}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{k=1}^{K} \text{sg}[\Omega_{i,k}] \left[1 - \cos\left(\mathbf{s}_{i,k}^{\text{int}}, \mathbf{s}_{i,k}^{\text{ext}}\right)\right]. \tag{15}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator. This objective encourages the planner's internal slots to align with external foresight, with the alignment strength modulated by guidance weights $\Omega_{i,k}$. Gradients through the internal slots update the shared planner representation, enabling training-time internalization.

The complete training objective is

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{planning}} + \lambda_{\text{internalization}} \mathcal{L}_{\text{internalization}} + \lambda_{\text{probe}} \mathcal{L}_{\text{probe}}, \tag{16}$$

where $\mathcal{L}_{\text{planning}}$ denotes the original planner training objective; $\lambda_{\text{internalization}}$ and λ_{probe} control foresight internalization and auxiliary probe supervision, respectively.

During training, LSDrive first establishes uniform foresight alignment through *Bootstrapping* and then activates hierarchical selection during *Refinement*. The stage-wise optimization schedule and gradient-flow details are provided in Appendix A.4. At inference, the external future-reasoning model, trajectory probe, and guidance-specific modules are removed. The resulting planner $f_{\theta^*}^{\text{planner}}$ predicts the ego trajectory directly from the current observation.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Datasets and metrics. We conduct experiments on nuScenes (Caesar et al., 2020) and NAVSIM (Dauner et al., 2024; Cao et al., 2025). nuScenes provides multi-view camera observations and future ego trajectories, and is widely used for open-loop trajectory-planning evaluation with L2 error and collision rate. NAVSIM evaluates planning in a non-reactive simulation environment using the PDM Score (PDMS), which aggregates multiple aspects of trajectory quality, including safety, driving progress, and passenger comfort.

Instantiation Settings. We instantiate LSDrive in two complementary settings based on representative future-aware driving frameworks. On nuScenes, FSDrive (Zeng et al., 2025) represents the explicit reasoning paradigm, in which a visual spatio-temporal CoT is autoregressively generated before ego-trajectory prediction. We reformulate its explicit future-reasoning pipeline within the LSDrive paradigm, enabling trajectory prediction without generating intermediate reasoning at inference. On NAVSIM, SGDrive (Li et al., 2026a) serves as the external foresight provider, supplying continuous world queries that encode future driving knowledge. We transfer future knowledge from SGDrive's world queries to ReCogDrive-based planner (Li et al., 2025d). Together, these settings assess whether LSDrive can internalize both explicit reasoning and latent future representations across planning architectures and datasets.

4.2 MAIN RESULTS ON NAVSIM

We evaluate LSDrive on NAVSIM from two complementary perspectives. First, we compare it with representative end-to-end, VLM/VLA-based, and future-aware driving methods under the official evaluation protocol to assess its overall planning performance. Second, we compare different strategies for utilizing external future evidence to evaluate the effectiveness of selective internalization.

Comparison with state-of-the-art methods. Table 1 compares LSDrive with representative planning methods on NAVSIM v1. Using a single front-view camera, LSDrive achieves a PDMS of 88.1 without additional post-training or inference-time trajectory refinement. It exceeds the SFT version of SGDrive by 0.7 points and matches the PDMS of DiffusionDrive, which uses three camera views and LiDAR. Importantly, LSDrive achieves this performance with a direct planner that requires neither the external world model nor its future representations at inference. These results demonstrate the practical viability of converting external foresight into training-time guidance.

LSDrive also benefits from additional post-training (91.0 PDMS) and inference refinement (89.8 PDMS). Combining both yields 91.8 PDMS, remaining highly competitive with the enhanced configurations of representative methods reported in the table. These results indicate that selective foresight internalization is compatible with further policy optimization and candidate trajectory selection. To further isolate the effect of selective internalization from differences in architecture and training settings, we compare alternative internalization strategies below.

Table 1: Comparison with representative methods on NAVSIM v1. Params denotes the total inference-time model size where available. Additional post-training and inference refinement are marked with ✓. NC: no at-fault collision. DAC: drivable area compliance. TTC: time-to-collision. C.: comfort. EP: ego progress. PDMS: the predictive driver model score. 1×C denotes single front-view camera. N×C denotes surround-view cameras. L denotes LiDAR. For LSDrive, post-training follows Li et al. (2025d) and inference refinement uses the best-of-N (N=6) strategy (Zhou et al., 2025).

Method	Sensors	Params	Post-training	Inference Refinement	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Human Driver	–	–	–	–	100.0	100.0	100.0	99.9	87.5	94.8
Conventional End-to-End Planners										
TransFuser (Chitta et al., 2023)	3×C + L	56M	–	–	97.7	92.8	92.8	100.0	79.2	84.0
PARA-Drive (Weng et al., 2024)	6×C	–	–	–	97.9	92.4	93.0	99.8	79.3	84.0
Hydra-MDP (Li et al., 2024)	3×C + L	–	–	✓	98.3	96.0	94.6	100.0	78.7	86.5
DiffusionDrive (Liao et al., 2025)	3×C + L	60M	–	–	98.2	96.2	94.7	100.0	82.2	88.1
VLM/VLA-based Planners										
AutoVLA (Zhou et al., 2025)	3×C	3B	✓	–	98.4	95.6	**98.0**	99.9	81.9	89.1
ReCogDrive (Li et al., 2025d)	3×C	8B	✓	–	98.2	97.8	95.2	99.8	83.5	89.6
Future-Aware Planners										
LAW (Li et al., 2025a)	1×C	–	–	–	96.4	95.4	88.7	99.9	81.7	84.6
DrivingGPT (Chen et al., 2024b)	1×C	–	–	–	98.9	90.7	94.9	95.6	79.7	82.4
Epona (Zhang et al., 2025)	1×C	2.5B	–	–	97.9	95.1	93.8	99.9	80.4	86.2
SGDrive (Li et al., 2026a)	1×C	2B	–	–	98.6	95.1	95.4	100.0	81.2	87.4
DriveWAM (Shi et al., 2026)	1×C	13B	–	–	98.3	98.1	95.2	100.0	84.3	90.1
SGDrive (Li et al., 2026a)	1×C	2B	✓	–	98.6	97.8	96.2	100.0	85.8	91.1
WoTE (Li et al., 2025c)	3×C + L	–	–	✓	98.5	96.8	94.9	99.9	81.9	88.3
DriveVLA-W0 (Li et al., 2026b)	1×C	8B	–	✓	98.7	**99.1**	95.3	99.3	83.3	90.2
DriveWorld-VLA (jia et al., 2026)	3×C	2B	✓	✓	**99.1**	98.2	96.1	100.0	85.9	91.3
LSDrive (Ours)	1×C	2B	–	–	98.6	96.2	95.2	100.0	81.8	88.1
LSDrive (Ours)	1×C	2B	✓	–	98.4	98.2	95.9	100.0	85.9	91.0
LSDrive (Ours)	1×C	2B	–	✓	98.8	97.3	95.9	100.0	83.6	89.8
LSDrive (Ours)	1×C	2B	✓	✓	98.7	98.5	97.5	100.0	**86.8**	**91.8**

Table 2: Comparison of future-knowledge learning strategies on NAVSIM v1. All variants use the same ReCogDrive planner, trajectory supervision, and Supervised Fine-Tuning (SFT) schedule. No variant uses reward-based post-training.

Method	Future Supervision Signal	Learning Strategy	PDMS ↑
ReCogDrive (Li et al., 2025d)	None	Trajectory supervision only	86.5
ReCogDrive + Future Prediction	Ground-truth future targets	Joint auxiliary prediction	87.1
ReCogDrive + Uniform Internalization	External-model foresight	Uniform latent alignment	87.5
ReCogDrive + LSDrive	External-model foresight	**Selective internalization**	**88.1**

Comparison of future evidence learning strategies. We compare four strategies for introducing future evidence into ReCogDrive (Li et al., 2025d). The first uses only trajectory-level SFT without introducing future evidence. The second jointly trains an auxiliary future-prediction task using ground-truth future supervision. The third employs the pretrained SGDrive world model as a frozen training-only provider and uniformly aligns all foresight slots. The fourth applies LSDrive to selectively internalize the same external foresight from the pretrained SGDrive world model at the scene and slot levels.

As shown in Table 2, uniform foresight distillation improves ReCogDrive from 86.5 to 87.5 PDMS, indicating that future knowledge acquired by the pretrained world model can be transferred into the direct planner. LSDrive further improves the performance to 88.1 PDMS, outperforming uniform transfer by 0.6 points and trajectory-only SFT by 1.6 points. This controlled comparison supports the effectiveness of allocating external guidance according to scene-level optimization priority and slot-level planning relevance. The future-prediction baseline further distinguishes external-model internalization from directly learning an auxiliary future task. Its comparison with uniform distillation evaluates the benefit of transferring knowledge from a separately pretrained world model, while its comparison with LSDrive determines whether planning-oriented evidence selection provides an advantage over dense future supervision.

Table 3: Planning performance and inference efficiency on nuScenes. FSDrive results are reproduced using its publicly released code. FSDrive generates explicit visual CoT at inference, whereas both internalization variants directly predict trajectories. Inference efficiency is measured using the same hardware and protocol.

Method	L2 Error (m)↓				Collision Rate (%)↓				Inference Efficiency			
	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.	Time (s/sample)↓	Throughput (samples/s)↑	Output Tokens per Sample↓	Visual-CoT Tokens per Sample↓
FSDrive (Explicit CoT)	0.48	1.12	1.95	1.18	0.07	0.34	1.38	0.60	0.3078	3.2575	481.50	383.75
Uniform Internalization	0.51	1.09	1.78	1.13	**0.02**	0.38	1.08	0.49	0.2123	4.7421	69.82	**0.00**
LSDrive (Ours)	**0.48**	**1.04**	**1.71**	**1.07**	0.07	**0.29**	**0.89**	**0.41**	**0.2101**	**4.7598**	**69.71**	**0.00**

4.3 Cross-Framework Validation on nuScenes

We further conduct experiments on nuScenes using FSDrive (Zeng et al., 2025), which generates an explicit visual chain-of-thought (CoT) before predicting the ego trajectory. We use the hidden states associated with its visual-CoT tokens as external future evidence and project them into foresight slots for internalization. Together with the experiments on NAVSIM, this setting evaluates LSDrive across different planning frameworks, datasets, and forms of future evidence.

Effectiveness of selective internalization. Table 3 compares FSDrive, uniform internalization, and LSDrive. Uniform internalization applies equal alignment weights to the projected foresight slots, while LSDrive weights their transfer through hierarchical foresight selection. LSDrive achieves an average L2 error of 1.07 m and an average collision rate of 0.41%, improving upon uniform internalization by 5.3% and 16.3%, respectively. It also outperforms the explicit-CoT planner, reducing these metrics by 9.3% and 31.7%. These results support the effectiveness of selective internalization for future evidence derived from explicit visual reasoning.

Inference efficiency. LSDrive eliminates visual-CoT generation at inference and reduces the number of output tokens per sample from 481.50 to 69.71. Under the measured protocol, generation time decreases from 0.3078 to 0.2101 seconds per sample, a reduction of 31.7%, while throughput increases by 46.1%. The improved efficiency accompanies lower trajectory error and collision rate, demonstrating that explicit visual-CoT rollout can be removed without sacrificing planning performance in this setting. Together with the NAVSIM results, these experiments support the applicability of the *Select-and-Internalize* paradigm across the two evaluated frameworks and datasets.

4.4 Ablation Studies

Effectiveness of Hierarchical Foresight Selection. Table 4 evaluates the contribution of each component in Hierarchical Foresight Selection. Compared with *Uniform Internalization*, which applies external foresight uniformly across scenes and slots, the full model reduces the L2 error from 1.13 to 1.07 and the collision metric from 0.49 to 0.41, showing that selectively controlling *when* and *which* foresight should guide the planner is more effective than indiscriminate transfer. Within Scene-Adaptive Guidance, removing Planning Difficulty (PD) or Latent Space Stability (LSS) degrades performance to 1.11/0.44 and 1.09/0.43, respectively, confirming that focusing on difficult scenes and suppressing mismatched foresight are complementary. For Planning-Relevant Foresight Scoring, removing Probe Supervision (P) causes the largest L2 degradation, while removing Slot-level Weighting (W) notably increases the collision metric to 0.47. This suggests that the probe branch improves the quality of relevance estimation, whereas Slot Weighting determines how this relevance is translated into selective guidance. Combining all components yields the best overall performance, further validating the complementary roles of scene-level guidance and slot-level relevance modeling in selective foresight internalization.

Slot intervention analysis. We compare three replacement references for Planning-Relevant Foresight Scoring. Each intervention replaces one future slot while keeping the planning context and all remaining slots unchanged. Zero replacement sets the selected feature vector to $\mathbf{0}$ without removing its position. Cross-sample shuffle instead uses the same-index slot from another sample, preserving the slot index while breaking its correspondence with the current scene. Batch mean uses a shared reference averaged over all samples and slots in the local batch, retaining aggregate feature statistics but discarding sample- and slot-specific information. As shown in Table 5, the zero-equivalent reference achieves the lowest L2 and collision metrics (1.07/0.41), followed by batch mean (1.11/0.46) and cross-sample shuffle (1.14/0.47). This is consistent with a sample-independent reference being preferable to introducing averaged or mismatched features when estimating slot importance. The results favor zero replacement for its simplicity and observed

Table 4: Ablation study of Hierarchical Foresight Selection. SAG denotes Scene-Adaptive Guidance, comprising Planning Difficulty (PD) and Latent Space Stability (LSS). PRFS denotes Planning-Relevant Foresight Scoring, which consists of the auxiliary probe-loss supervision (P) and slot-level weighting (W).

Variant	SAG		PRFS		L2 Avg ↓	Col. Avg ↓
	PD	LSS	P	W		
Uniform Internalization	×	×	×	×	1.13	0.49
w/o Planning Difficulty	×	✓	✓	✓	1.11	0.44
w/o Latent Space Stability	✓	×	✓	✓	1.09	0.43
w/o Probe Supervision	✓	✓	×	✓	1.13	0.46
w/o Slot Weighting	✓	✓	✓	×	1.10	0.47
Full Model	✓	✓	✓	✓	**1.07**	**0.41**

Table 5: Slot intervention ablation for training-time foresight scoring. Each intervention replaces one slot while preserving slot count and dimensionality. Zero sets the selected vector to $\mathbf{0}$; cross-sample shuffle uses $f_{\pi(i),k}$, where $\pi(i)$ indexes another sample in the local batch. Batch mean uses $\bar{f} = \frac{1}{NK}\sum_{j=1}^{N}\sum_{l=1}^{K} f_{j,l}$, where N is the local batch size and K is the number of slots.

Replacement	$f'_{i,k}$	L2 Avg ↓	Col. Avg ↓
Zero	$\mathbf{0}$	**1.07**	**0.41**
Cross-sample shuffle	$f_{\pi(i),k}$	1.14	0.47
Batch mean	$\bar{f}$	1.11	0.46

Table 6: Trajectory prediction accuracy and safety metrics of the Planner versus the auxiliary Probe.

Metric	**Planner**	**Probe**
L2 @ 1s (m) ↓	0.48	0.61
L2 @ 2s (m) ↓	1.04	1.17
L2 @ 3s (m) ↓	1.71	1.85
L2 Avg. (m) ↓	1.07	1.21
Collision Avg. (%) ↓	0.41	0.59

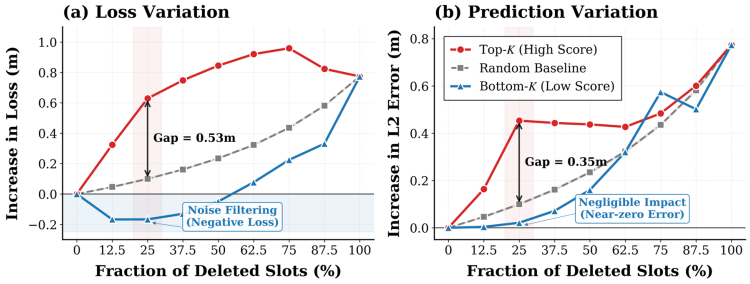

Figure 3: Impact of slot deletion on Loss and Prediction variations.

planning performance. More detailed analyses of specific internalization strategies, training schedule optimization, and hyperparameter sensitivity are provided in Appendix B.

4.5 Mechanism Analysis

Probe Accuracy and Slot Relevance. As shown in Table 6, the auxiliary probe achieves an average L2 error of 1.21 m and a collision rate of 0.59%, maintaining a marginal gap (∼0.14 m) with the deployed planner. This confirms that the probe retains sufficient predictive capability to serve as a reliable evaluator, without needing to match the planner's absolute performance. More importantly, we verify whether the probe genuinely discriminates decision-critical foresight. As illustrated in Fig. 3, the deletion analysis yields two key conclusions. **(1) Criticality of high-scoring slots.** Removing top-ranked slots (Top-K) causes a precipitous spike in both loss and prediction error, yielding a massive gap against the Random baseline (e.g., 0.53 m in loss at 25% deletion). This proves these slots carry indispensable semantic information. **(2) Noise-filtering effect of low-scoring slots.** Conversely, removing bottom-ranked slots (Bottom-K) results in a near-zero prediction error increase and even a *negative loss increment*. This demonstrates that low-scoring slots actively disrupt optimization, rather than being merely redundant. These results validate the effectiveness of our scoring mechanism, providing reliable guidance for the selective internalization into the planner. Furthermore, a quantitative analysis regarding the Latent Space Stability are deferred to Appendix B.

5 Conclusion

We presented LatentSightDrive (LSDrive), a Select-and-Internalize framework that selectively transfers external foresight into a direct driving planner. Through a common foresight-slot interface and hierarchical scene- and slot-level selection, LSDrive progressively shapes the planner's latent representation with planning-relevant future evidence. The resulting planner operates without the external world model or auxiliary guidance modules at inference. Experiments on NAVSIM and nuScenes demonstrate the effectiveness of selective internalization across latent world representations and explicit visual-CoT reasoning, improving planning performance over uniform transfer while eliminating deployment-time dependence on external future reasoning.

The Use of Large Language Models (LLMs)

Large Language Models were used as a writing assistance tool in this work. Their role was strictly limited to grammar checking, sentence polishing, and improving the fluency and clarity of the manuscript's language. The models did not contribute to research ideas, experimental design, and data analysis. All technical results, claims, and conclusions remain the sole responsibility of the authors.

Ethics Statement

This work complies with the ICLR Code of Ethics. No ethical concerns requiring specific disclosure are present.

Reproducibility Statement

To ensure reproducibility, we provide: (1) implementation details; (2) the training protocols follow the works (Li et al., 2026a; Zeng et al., 2025; Li et al., 2025d), of which the codes are all released; (3) the training and inference code upon paper acceptance.

References

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 (CVPR)*, June 2020.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024a.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers, 2024b. URL https://arxiv.org/abs/2412.18607.

Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Ppad: Iterative interactions of prediction and planning for end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 239–256. Springer, 2024c.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700, 2018. doi: 10.1109/ICRA.2018.8460487.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation, 2023. URL https://arxiv.org/abs/2311.01460.

Kaituo Feng, Changsheng Li, Dongchun Ren, Ye Yuan, and Guoren Wang. On the road to portability: Compressing end-to-end motion planner for autonomous driving, 2024. URL https://arxiv.org/abs/2403.01238.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. 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.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

Feiyang jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving, 2026. URL https://arxiv.org/abs/2602.06521.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316, 2023. doi: 10.1109/ICCV51070.2023.00766.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

Guangfeng Jiang, Yueru Luo, Jun Liu, Yi Huang, Yiyao Zhu, Zhan Qu, Dave Zhenyu Chen, Bingbing Liu, and Xu Yan. Wpt: World-to-policy transfer via online world model distillation, 2026. URL https://arxiv.org/abs/2511.20095.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. *arXiv preprint arXiv:2601.05640*, 2026a.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=fd2u60ryG0.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model, 2025b. URL https://arxiv.org/abs/2406.08481.

Yingyan Li, Yu-Quan Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146, 2025c. URL https://api.semanticscholar.org/CorpusID:277501699.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, AnYasong, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-w0: World models amplify data scaling law in autonomous driving. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=plrGn3RdzN.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving, 2025d. URL https://arxiv.org/abs/2506.08052.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation, 2024. URL https://arxiv.org/abs/2406.06978.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025. doi: 10.1109/CVPR52734.2025.01124.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. Jepa-wam: Learning vision-language-action policies with joint-embedding world modeling, 2026. URL https://arxiv.org/abs/2608.09381.

Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, Liping Jing, Yiming Nie, and Bin Dai. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15522–15533, 2024. doi: 10.1109/CVPR52733.2024.01470.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. 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.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation, 2025. URL https://arxiv.org/abs/2502.21074.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. Drivewam: Video generative priors enable scalable world-action modeling for autonomous driving, 2026. URL https://arxiv.org/abs/2605.28544.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. In *Conference on Robot Learning*, pp. 4698–4726. PMLR, 2025.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024. doi: 10.1109/CVPR52733.2024.01397.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15449–15458, 2024. doi: 10.1109/CVPR52733.2024.01463.

Jiwei Yang, Zhengxian Chen, Chaosheng Huang, and Jun Li. Auto-jepa: A latent world model of continuous intent for end-to-end autonomous driving, 2026. URL https://arxiv.org/abs/2607.29031.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=fCirUh6FRb.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, XianPeng Lang, and Dongbin Zhao. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, 2025. doi: 10.1109/ICCV51701.2025.02659.

Zewei Zhou, Tianhui Cai, Yun Zhao, Seth Z.and Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Additional Method Details

This appendix provides implementation and optimization details complementary to Sec. 3. We describe foresight-slot projection, the running foresight reference, trajectory-probe implementation, and stage-wise training and inference configurations.

A.1 Foresight Slot Projection Details

Planner-side foresight projection. The planner-side projector first maps $\mathbf{H}_i^{\text{planner}}$ to the common feature dimension $d_{\text{foresight}}$ using a trainable feature adapter. The learnable queries $\mathbf{Q}^{\text{future}} \in \mathbb{R}^{K \times d_{\text{foresight}}}$ then cross-attend to the adapted representation:

$$\mathbf{S}_i^{\text{int}} = \text{QueryAggregate}_{\text{int}}\left(\mathbf{Q}^{\text{future}}, A_{\text{int}}(\mathbf{H}_i^{\text{planner}})\right). \tag{17}$$

Here, A_{int} denotes the feature adapter, and $\text{QueryAggregate}_{\text{int}}$ denotes cross-attention followed by residual feed-forward processing. The output contains K sample-conditioned slots ordered by their query indices.

External foresight projection. The external projector uses a separate feature adapter and an independent set of learnable queries to aggregate external future evidence:

$$\mathbf{S}_i^{\text{ext}} = \text{QueryAggregate}_{\text{ext}}\left(\mathbf{Q}^{\text{external}}, A_{\text{ext}}(\mathbf{E}_i)\right), \qquad \mathbf{Q}^{\text{external}} \in \mathbb{R}^{K \times d_{\text{foresight}}}. \tag{18}$$

The external queries, feature adapter, and aggregation parameters are collectively included in ψ_{ext} in the main text. This learnable aggregation converts variable-length external evidence into K slots matching the internal representation in both slot number and feature dimension.

Slot correspondence. The internal and external projectors use independent query sets, each of which defines a fixed output ordering. The alignment objective explicitly pairs internal slot k with external slot k; it is therefore an index-based pairing rather than an order-invariant distribution-matching objective.

The paired indices do not impose predefined semantic categories such as lanes or agents. During *Bootstrapping*, uniform alignment encourages each internal slot to match its indexed external counterpart, allowing slot-wise semantic correspondence to emerge through training rather than being assumed at initialization. *Refinement* preserves the same pairing while modulating its alignment strength through $\Omega_{i,k}$. Consequently, the planning relevance estimated for external slot k controls its transfer to internal slot k.

A.2 Scene-Adaptive Guidance Details

Running foresight reference. We maintain an exponential-moving-average reference of the planner-side foresight representation:

$$\overline{\mathbf{S}}_t^{\text{int}} = \beta \overline{\mathbf{S}}_{t-1}^{\text{int}} + (1-\beta)\,\text{sg}\left[\frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \mathbf{S}_i^{\text{int}}\right], \tag{19}$$

where β controls temporal smoothing and $\text{sg}[\cdot]$ denotes stop-gradient. This reference provides the adaptive scale used by d_i^{ref} in Eq. 7.

Running-reference implementation. The running reference is maintained as a non-trainable buffer. It is initialized with the mean planner-side foresight representation of the first mini-batch and subsequently updated using an exponential moving average with $\beta = 0.99$. The update is performed whenever the internalization objective is evaluated while the external future-reasoning model is active. At each iteration, Scene-Adaptive Guidance uses the reference from the previous update before incorporating statistics from the current mini-batch. The update uses local mini-batch statistics without cross-device synchronization, and the reference is maintained entirely without gradient tracking.

A.3 Trajectory Probe and Slot Intervention

Planning-Relevant Foresight Scoring uses a lightweight trajectory probe only during training. The probe receives the projected external foresight slots together with planner-side driving context and predicts the future ego trajectory.

For slot intervention, zero masking preserves both sequence length and slot index while removing the information carried by the selected slot:

$$\mathbf{S}_{i,\setminus k}^{\text{ext}} = \left[\mathbf{s}_{i,1}^{\text{ext}}, \ldots, \mathbf{0}, \ldots, \mathbf{s}_{i,K}^{\text{ext}}\right]^\top, \tag{20}$$

where the k-th slot is replaced by a zero vector. The corresponding trajectory degradation provides the relevance score in Eq. 13.

Probe architecture. The planning probe is a lightweight three-layer MLP trajectory decoder with two 256-dimensional hidden layers, GELU activations, and LayerNorm after the first linear layer. Its input concatenates the external foresight slots with a planner-side representation encoding the current visual context, ego-motion history, and navigation command. The decoder predicts an ego trajectory in the same format as the ground-truth trajectory and is trained using mean absolute trajectory error. The same probe is used for both full-slot and leave-one-slot-out predictions, while the planner-side conditioning remains unchanged.

Probe gradient flow. The probe loss $\mathcal{L}_{\text{probe}}$ updates the trajectory probe q_ϕ^{probe} and the external slot projector $\Pi_{\psi_{\text{ext}}}$, while gradients are stopped before reaching the planner backbone and the frozen external future-reasoning model.

Slot intervention. For each leave-one-slot-out intervention, the selected external foresight slot is replaced by a zero vector while its position and all remaining inputs are kept unchanged. Table 5 compares this zero replacement with cross-sample shuffle and batch-mean replacement.

A.4 Training and Inference Details

Stage-wise training optimization. Training consists of two stages: *Bootstrapping* and *Refinement*. Bootstrapping establishes an initial alignment between planner-side and external foresight slots before hierarchical selection is activated.

During Bootstrapping, all scenes receive equal guidance, and the internalization loss, is averaged uniformly over the foresight slots:

$$\Omega_{i,k}^{\text{uni}} = \frac{1}{K}, \qquad \forall i, k. \tag{21}$$

During Refinement, the uniform weights are replaced by

$$\Omega_{i,k} = [\mathcal{G}]_i [\mathcal{S}]_{i,k}, \tag{22}$$

activating scene- and slot-level selective internalization.

Training schedule. On NAVSIM, training runs for 100 epochs. Bootstrapping occupies epochs 0-19, followed by Refinement from epoch 20 onward. Scene-level guidance and probe supervision are activated at epoch 20, while slot-level weighting starts at epoch 25 after five epochs of probe warm-up.

On nuScenes, we follow the same scheduling principle over a 12-epoch budget (epochs 0-11). Bootstrapping occupies epochs 0-1, Refinement spans epochs 2-11, and slot-level weighting is activated starting at epoch 3, following one epoch of probe warm-up.

Gradient flow. The external future-reasoning model remains frozen throughout training. The internalization loss $\mathcal{L}_{\text{internalization}}$ updates the planner backbone, the planner-side slot projector through $\mathbf{S}_i^{\text{int}}$, and the external foresight target $\mathbf{S}_i^{\text{ext}}$. The hierarchical guidance weights $\mathbf{\Omega}$ are treated as stop-gradient quantities in this objective.

The probe loss $\mathcal{L}_{\text{probe}}$ separately updates the trajectory probe and the external slot projector, without propagating gradients into the planner backbone or the frozen external future-reasoning model. At each iteration, $\mathbf{\Omega}$ is recomputed from the current planner representations and probe evaluations, detached for weighting the current internalization update, and recomputed again at the next iteration.

Inference graph. At inference, the external world model, both slot-projection branches, Hierarchical Foresight Selection, and the trajectory probe are removed, leaving only the original observation-to-trajectory planning pathway. The direct planner predicts the ego trajectory from $\mathbf{H}_i^{\text{planner}}$, which is extracted from the current driving observation and context.

Although all auxiliary branches are removed, their training influence is retained in the planner backbone. Selective foresight internalization backpropagates planning-relevant future supervision into the planner parameters; these updated parameters subsequently produce $\mathbf{H}_i^{\text{planner}}$ at inference. Future knowledge therefore affects deployment-time planning through the learned parameters and resulting representations rather than through an additional future-reasoning input.

Training overhead. We report training costs under the current experimental configurations. On nuScenes, the FSDrive-based direct planner uses *cached* visual-CoT tokens and requires approximately 4.5 GPU-hours with 30 GB

peak GPU memory for training. LSDrive loads the frozen external teacher during training and requires approximately 5.5 GPU-hours with 125 GB peak GPU memory. On NAVSIM, ReCogDrive serves as the direct planner and SGDrive provides external foresight. The ReCogDrive baseline requires approximately 30 GPU-hours with 160 GB peak GPU memory, whereas LSDrive requires approximately 70 GPU-hours with a similar peak memory footprint. At inference, the external teacher and all guidance-specific branches are removed, leaving the same direct planning architecture without additional inference-time modules or computation.

B Additional Ablation Studies and Analysis

In this appendix, we provide comprehensive ablation studies and analysis to validate the design choices of LSDrive. The experiments are organized into four key aspects: internalization strategies, training schedule optimization, hyperparameter sensitivity, and latent space stability.

B.1 Internalization Strategies

Table 7 compares different strategies for internalizing external foresight into the planner. **LSDrive** significantly outperforms the *Uniform*, *Random*, and *Attention* baselines. This demonstrates that blindly injecting future information (Uniform) or relying on implicit attention mechanisms is suboptimal. Our explicit, planning-relevance-guided selective mechanism identifies and distills decision-critical future evidence, yielding the lowest L2 error and collision rate.

Table 7: Ablation study on foresight internalization strategies. **LSDrive** outperforms uniform, random, and attention-based baselines, demonstrating the effectiveness of our selective mechanism.

Method	**L2 Error (m) ↓**				**Collision Rate (%) ↓**			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
FSDrive (Explicit CoT)	0.48	1.12	1.95	1.18	0.07	0.34	1.38	0.60
Random Internalization	0.52	1.11	1.81	1.15	0.02	0.44	1.11	0.52
Uniform Internalization	0.51	1.09	1.78	1.13	**0.02**	0.38	1.08	0.49
Attention Internalization	0.50	1.07	1.74	1.10	0.05	0.39	1.06	0.50
LSDrive (Ours)	**0.48**	**1.04**	**1.71**	**1.07**	0.07	**0.29**	**0.89**	**0.41**

B.2 Training Schedule Optimization

The progressive internalization relies on a two-stage training schedule: Stage-1 (Bootstrapping) establishes initial alignment, and Stage-2 (Refinement) performs selective distillation. Table 8 compares different epoch allocations under a fixed 12-epoch budget on nuScenes.

The results reveal a clear sensitivity to the stage allocation. Extending the bootstrapping phase (e.g., B4-R8) degrades the performance to 1.14 m L2 and 0.51% collision. This indicates that excessive uniform bootstrapping forces the planner to align with redundant or noisy external slots, which over-smooths the latent representations and subsequently hinders the effectiveness of the selective refinement stage. Conversely, the **B2-R10** configuration strikes the optimal balance: it provides a sufficiently stable initialization for the planner while maximizing the training budget for the crucial selective refinement process.

B.3 Hyperparameter Sensitivity

Table 9 presents the one-factor-at-a-time analysis for two key parameters: probe-loss coefficient λ_{probe} and hard-case quantile q. The marked values (†) indicate the final selected parameters.

- **Probe-loss coefficient (λ_{probe}):** A moderate value of 0.025 yields the best robustness. A smaller weight (0.015) under-regularizes the probe, while a larger weight (0.050) over-constrains the probe's optimization, both leading to performance degradation.
- **Hard-case quantile (q):** The quantile $q = 0.70$ (selecting the top 30% hardest samples) achieves the optimal balance. A stricter threshold ($q = 0.75$) reduces the sample size too much, limiting the refinement signal, while a looser threshold ($q = 0.65$) includes easier samples that dilute the selective distillation.

Table 8: Training schedule comparison.

Config.	**B**ootstrapping	**R**efinement	**L2 Avg** ↓	**Col. Avg** ↓
B2-R10	2	10	**1.07**	**0.41**
B3-R9	3	9	1.10	0.47
B4-R8	4	8	1.14	0.51
B5-R7	5	7	1.12	0.45
B6-R6	6	6	1.12	0.47

Table 9: Hyperparameter sensitivity analysis. † indicates the final selected parameters.

Parameter	**Value**	**L2 Avg** ↓	**Col. Avg** ↓
λ_{probe}	0.015	1.15	0.49
	0.025†	**1.07**	**0.41**
	0.050	1.09	0.43
Hard-case quantile q	0.65	1.09	0.46
	0.70†	**1.07**	**0.41**
	0.75	1.11	0.48

B.4 Analysis of Latent Space Stability

To prevent disruptive alignment during internalization, we introduce a scene-adaptive weighting mechanism, Latent Space Stability (LSS), to maintain the internalization stability. Table 10 validates its necessity by comparing three weighting strategies for hard samples:

- **w/o LSS:** Assigns equal weights to all hard samples, ignoring representation discrepancy.
- **Discrepancy-Amplified:** Assigns *higher* weights to samples with larger external-internal discrepancies.
- **LSS (Ours):** Assigns *lower* weights to larger discrepancies, acting as a safeguard to maintain stability.

Table 10: Ablation on scene-adaptive weighting strategies for Latent Space Stability.

Setting	**Weighting Strategy**	**L2 Avg. (m)** ↓	**Collision Avg. (%)** ↓
w/o LSS	Equal weights for hard samples	1.09	0.43
Discrepancy-Amplified	Higher weight for larger discrepancy	1.12	0.50
LSS (Ours)	**Lower weight for larger discrepancy**	**1.07**	**0.41**

Table 10 yields key mechanistic insights: (1) Amplifying discrepancies is detrimental, as the *Discrepancy-Amplified* setting yields the worst performance (1.12 m L2, 0.50% collision), empirically suggesting that forcing strong alignment on high-discrepancy samples may introduce noise that disrupts the planner's ongoing optimization; (2) Attenuating discrepancies ensures safety, as our LSS mechanism improves both L2 error (1.09 m $\rightarrow$ 1.07 m) and collision rate (0.43% $\rightarrow$ 0.41%) compared to the *w/o LSS* baseline, demonstrating that moderating alignment strength in ambiguous scenes effectively prevents the planner from being misled by less reliable external cues; (3) Overall, these results validate that LSS acts as a necessary safeguard, striking an optimal balance by allowing the planner to learn from consistent external foresight while preserving its stable representation when external cues are highly divergent.