DUALWM: DUAL-LAYER UNIFIED ASYNCHRONOUS LATENT WORLD MODEL FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models enhance autonomous driving by learning task-state transitions, providing valuable policy priors. However, existing methods suffer from the fundamental conflict between cognitive reasoning and real-time responsiveness in dynamic driving scenarios. To address this challenge, we propose a novel end-to-end autonomous driving framework based on a Dual-layer Unified Asynchronous Latent World Model (DualWM) that learns state transitions at both the intent and behavioral levels within a unified latent space for efficient driving. Specifically, the VLM-based upper layer handles scene understanding and long-horizon intent, while the lower layer enables real-time semantic state prediction. The two layers are mutually coupled: the intent features provide long-horizon priors that guide behavioral prediction, and the behavioral states in turn constrain the uncertainty inherent in VLM predictions, while the asynchronous architecture improves inference efficiency. This design draws inspiration from human driving behavior, where conscious planning coexists with rapid reactive adjustments. Finally, a Mixture-of-Experts (MoE)-based policy module is designed to capture both the consistency and conflicts between macroscopic intent and behavioral-level decisions, enabling efficient and robust closed-loop driving. Extensive experiments on the Bench2Drive closed-loop benchmark in CARLA demonstrate that DualWM achieves strong closed-loop driving performance while substantially improving runtime efficiency, validating its potential for real-world autonomous driving.

1 INTRODUCTION

Autonomous driving has emerged as a major research focus across academia and industry, with the potential to improve road safety and transportation efficiency. Early systems followed a modular pipeline that decomposed the driving task into perception, motion prediction, and planning (Paden et al., 2016; Chen et al., 2024a). To mitigate error propagation and information loss across manually designed module interfaces, subsequent work has increasingly explored end-to-end (E2E) approaches that map sensor inputs to driving controls or trajectories within a unified framework (Codevilla et al., 2018; Chitta et al., 2023; Hu et al., 2022b), as illustrated in Figure 1(a). However, robust closed-loop performance in long-tail scenarios involving complex multi-agent interactions remains challenging for autonomous driving systems (Jia et al., 2024).

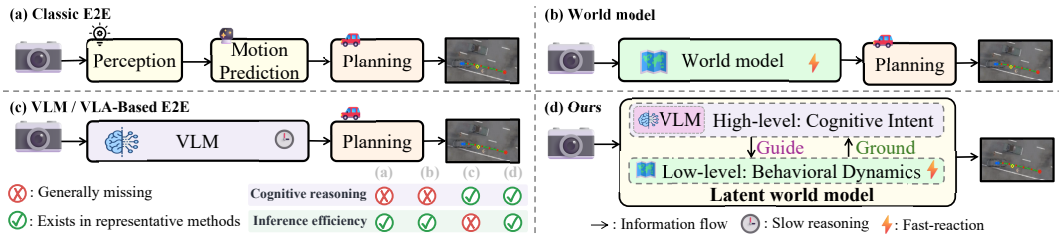


Figure 1: Comparing E2E driving paradigms. DualWM couples VLM-derived intent with behavioral dynamics in a latent world model, achieving cognitive reasoning and efficiency.

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.

To address this challenge, world models offer a promising avenue by learning rich representations of scene geometry and transition dynamics, enabling anticipation of scene evolution and adaptation to unfamiliar conditions. Existing methods broadly fall into two categories: pixel-space models and latent world models (LWMs). The former synthesize future observations for simulation but at high computational cost (Wang et al., 2024b;a), whereas the latter learn dynamics over compact latent representations, avoiding expensive pixel-level generation (Zhang et al., 2024; Min et al., 2024; Li et al., 2025a; Yang et al., 2025). The predictive priors that LWMs provide for policy learning make these models attractive for computationally constrained closed-loop driving. While most LWMs excel at learning motion-level physical dynamics, they remain less developed in high-level cognitive reasoning to guide policy learning in complex traffic scenes, as illustrated in Figure 1(b).

The integration of Vision Language Models (VLMs) significantly enhances the high-level cognitive capabilities of latent world models for autonomous driving. These capabilities have motivated the growing integration of VLMs into E2E driving systems, as illustrated in Figure 1(c). However, autoregressive VLM inference incurs substantial computational overhead, making real-time deployment difficult (Liu et al., 2025; Hu et al., 2025). Existing approaches reduce VLM inference cost through distillation into lightweight networks (Liu et al., 2025) or fast-slow architectures that retain large-model guidance online (Chen et al., 2024b; Tian et al., 2025; Zhang et al., 2025; Hamdan et al., 2025). Such designs improve computational efficiency, while reliable driving also requires grounding maneuver-level intent in scene dynamics. This is particularly important because spatial perception errors and hallucinations can compromise the reliability of the guidance produced by VLMs (Huang et al., 2026). A few recent studies have attempted to combine VLMs with world models (Mei et al., 2026; Xiong et al., 2026; Tan et al., 2026). However, maintaining alignment between cognitive intent and evolving behavioral dynamics remains challenging.

To address these challenges, we propose a novel E2E autonomous driving framework centered on a **D**ual-layer **U**nified **A**synchronous **L**atent **W**orld **M**odel (DualWM), as illustrated in Figure 1(d). DualWM integrates VLM-based high-level cognition with efficient behavioral dynamics modeling to reconcile comprehensive scene understanding with responsiveness in complex traffic environments. Inspired by human driving, where deliberate reasoning operates alongside fast reflexive control, DualWM comprises three key designs: (i) We introduce a dual-layer latent world model with coupling between VLM-based intent and behavioral dynamics to guide and ground decisions. (ii) We develop an asynchronous architecture that updates both layers at different rates, since cognitive intent evolves more slowly than vehicle dynamics, thereby amortizing VLM inference across control steps. (iii) We design an MoE-based policy module that routes intent-driven representations to behavior-specific experts, enabling the planner to balance routine and expert behaviors for different maneuvers. Our contributions are summarized as follows:

- We propose a unified cognitive-behavioral end-to-end autonomous driving framework that captures intent dynamics at the cognitive level, while learning motion transitions at the behavioral level. In addition, an MoE-based planner is elaborated to reconcile intent-behavior conflicts and consistencies for stable, precise policy generation.
- A novel DualWM is proposed where an asynchronous dual-layer probabilistic graphical model is constructed to improve the deployability of VLMs at a lower inference cost in highly dynamic driving tasks. The learned behavioral state transitions serve as constraints to mitigate the uncertainty and interpretability issues inherent in VLMs.
- Extensive experimental results demonstrate that, compared with other VLM-based driving methods, DualWM achieves superior driving performance with substantially lower inference latency, validating its dual advantages in inference efficiency and driving capability.

2 Related Work

2.1 End-to-End Autonomous Driving

End-to-end autonomous driving maps sensor observations directly to driving plans within a unified learning framework (Wu et al., 2022; Jia et al., 2023a;b). BEVFormer (Li et al., 2022) strengthens temporal BEV feature aggregation, while UniAD (Hu et al., 2023b) and VAD (Jiang et al., 2023) organize perception, prediction, and planning into unified planning-oriented architectures. Recent planners further diversify trajectory generation: VADv2 (Jiang et al., 2026) models distributions over

discretized actions, Hydra-MDP (Li et al., 2024) distills multiple teachers into a multi-head planner, GenAD (Zheng et al., 2024b) formulates joint prediction and planning as generative sampling, and DriveTransformer (Jia et al., 2025) uses sparse queries to attend directly to sensor features. These architectures provide a strong foundation for closed-loop driving policy learning.

2.2 World Models for Autonomous Driving

World models provide critical predictive priors for autonomous driving policies by learning the transition regularities of navigation states. Explicit world models can directly generate future driving videos, yet their high computational cost makes them unsuitable for real-time navigation tasks (Hu et al., 2023a; Gao et al., 2024). LWMs offer an efficient and real-time paradigm for capturing navigation dynamics, delivering critical foresight that strengthens autonomous driving in traffic environments (Hu et al., 2022a; Zheng et al., 2024a; Li et al., 2025a). World4Drive (Zheng et al., 2025) incorporates intention-aware priors from vision foundation models. WoTE (Li et al., 2025b) further rolls out future BEV states for multiple trajectory proposals and scores them with a learned reward model. However, these works still fall short of adequately capturing the interaction dynamics between vehicles and their surrounding environment at the cognitive level.

2.3 Vision Language Models in Autonomous Driving

VLMs have recently been introduced into autonomous driving to provide intention understanding and commonsense reasoning (Fu et al., 2026; Li et al., 2026; Yang et al., 2026). DriveVLM (Tian et al., 2025) derives hierarchical decisions through chain-of-thought reasoning, ORION (Fu et al., 2025) maps VLM reasoning to planning tokens, and DriveLM (Sima et al., 2024) improves interpretability through language supervision. Fast-slow architectures retain large-model guidance during online planning while reducing its invocation frequency (Chen et al., 2024b; Tian et al., 2025; Zhang et al., 2025; Hamdan et al., 2025). Recent studies also combine VLMs with world models to model scene dynamics (Mei et al., 2026; Xiong et al., 2026; Tan et al., 2026). We focus on the interaction between intent-level and behavioral dynamics: intent guides behavioral prediction, and behavioral evidence informs cognitive updates. Our method introduces a dual-layer asynchronous architecture that utilizes the cognitive reasoning of VLMs during navigation while mitigating the inference latency inherent to large models.

3 Method

In this paper, we propose DualWM, an E2E autonomous driving framework comprising an asynchronous dual-layer latent world model and a cognition-behavior coupled planner, as shown in Figure 2. The former comprises a slow cognitive intent layer that transforms frozen VLM features into persistent maneuver-level intent and a fast behavioral dynamics layer that models intent-conditioned BEV dynamics from frozen BEV representations. Through bidirectional asynchronous coupling, high-level intent guides behavioral prediction, while physically grounded dynamics constrain VLM-derived cognitive uncertainty. The latter grounds cognitive intent in behavioral dynamics and augments a base trajectory with intent-gated expert residuals to generate the ego trajectory, thereby enabling safe and efficient driving.

3.1 Dual-Layer Unified Latent World Model

To integrate VLM-based cognition with physically grounded world modeling, we formulate DualWM as a dual-layer latent world model comprising a Cognitive Intent Layer and a Behavioral Dynamics Layer, coupled through the joint state $\mathbf{s}_t = [\bar{\mathbf{s}}_t^h; \mathbf{s}_t^l]$ for future prediction and planning. Control frames are indexed by $t = 0, \dots, T$, with the VLM queried every K frames on $\mathcal{T}_c = \{0, K, 2K, \dots\}$ and $\kappa(t) = K\lfloor t/K \rfloor$ denoting the latest query frame. The cognitive state is updated only on $\mathcal{T}_c$ and held between queries as $\bar{\mathbf{s}}_t^h = \mathbf{s}_{\kappa(t)}^h$, whereas the behavioral state $\mathbf{s}_t^l$ evolves at every frame. For each query frame $t \in \mathcal{T}_c$, the subsequent actions are grouped into the chunk $\mathbf{u}_t = (\mathbf{a}_t, \dots, \mathbf{a}_{t+K-1})$.

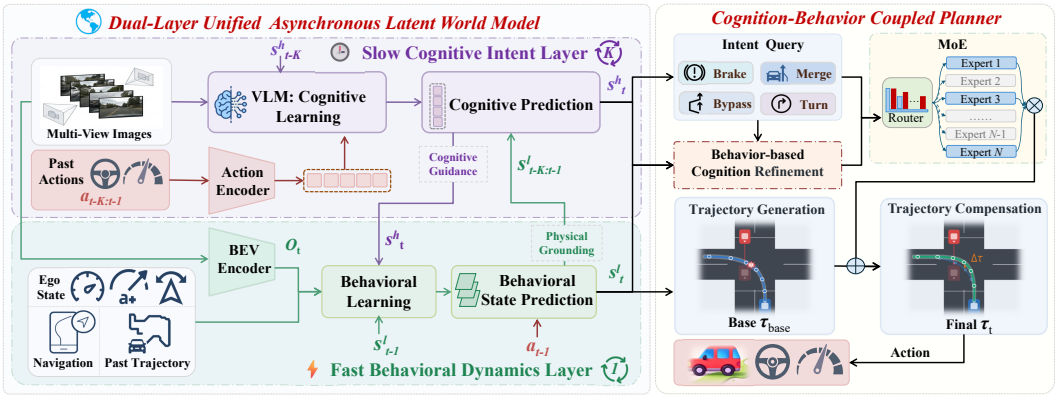


Figure 2: Overview of DualWM, an E2E autonomous driving framework comprising an asynchronous dual-layer latent world model and a cognition-behavior coupled planner. The model couples slow VLM intent reasoning with fast BEV dynamics, while the planner grounds cognitive intent in behavioral dynamics and generates ego trajectories via intent-gated residual experts.

3.1.1 Cognitive Intent Layer.

The Cognitive Intent Layer is pivotal in DualWM because complex driving scenarios require high-level intent reasoning beyond low-level scene dynamics. We employ a frozen VLM as the cognitive backbone to extract semantic and commonsense cues from multi-view images. On each query frame $t \in \mathcal{T}_c$, the VLM produces a cognitive feature $\mathbf{c}_t$, which is absorbed into the recurrent latent state $\mathbf{s}_t^h$, and an intent head decodes lateral, longitudinal, and safety-related intentions from this state. The cognitive state thus encodes the ego vehicle's committed maneuver rather than the current sensory scene. Persisting across the query interval, it provides a long-horizon prior that the behavioral layer relies on between VLM queries.

3.1.2 Behavioral Dynamics Layer.

The Behavioral Dynamics Layer complements the Cognitive Intent Layer by modeling physical scene evolution under ego motion and surrounding-agent interactions. We therefore formulate the lower layer as a BEV dynamics model conditioned on the current observation, previous ego action, and held cognitive state. Given surround-view images at frame t, a frozen BEV encoder produces scene observation $\mathbf{o}_t$, which is combined with the previous action $\mathbf{a}_{t-1}$ and the held cognitive state $\bar{\mathbf{s}}_t^h$ to update the recurrent behavioral state $\mathbf{s}_t^l$. From this state, a reconstruction head recovers the current BEV representation, while a rollout head predicts future BEV states over the planning horizon. Conditioning the transition on $\bar{\mathbf{s}}_t^h$ makes the rollout intent-aware, allowing the same observation to yield distinct scene evolutions under different maneuver-level intents.

3.2 Asynchronous Cross-Layer Coupling Mechanism

Invoking the VLM at every control step is computationally costly and often unnecessary, because high-level driving intent evolves more slowly than local vehicle dynamics. However, naively lowering the VLM query frequency risks making the cognitive signal stale or inconsistent with the behavior executed by the vehicle. We therefore introduce an asynchronous cross-layer coupling mechanism that preserves cognitive guidance between VLM queries and grounds each cognitive update in the vehicle's executed motion. We formalize this mechanism as a dual-layer sequential latent-variable model, whose dependency structure is shown in Figure 3. Write $S = (\mathbf{s}_{\mathcal{T}_c}^h, \mathbf{s}_{0:T}^l)$ for the latent trajectory and $X = (\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c})$ for the observed targets, where $\mathbf{y}_t$ denotes the maneuver annotations. The stochastic components of the latent states follow diagonal Gaussian distributions parameterized by neural networks, with generative parameters θ and inference parameters ϕ. The generative model factorizes over the two temporal scales as

$$p_\theta(X, S \mid \mathbf{a}) = \prod_{t \in \mathcal{T}_c} p_\theta(\mathbf{s}_t^h \mid \mathbf{s}_{t-K}^h, \mathbf{u}_{t-K}, \mathbf{s}_{t-K:t-1}^l)\, p_\theta(\mathbf{y}_t \mid \mathbf{s}_t^h) \prod_{t=0}^{T} p_\theta(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h)\, p_\theta(\mathbf{o}_t \mid \mathbf{s}_t^l), \tag{1}$$

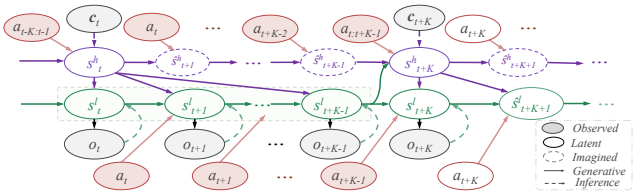


Figure 3: Our dual-layer probabilistic graphical model.

where $\mathbf{a}$ is the executed action sequence and $\mathbf{u}$ its per-interval chunks. The cognitive transition advances every query interval, the behavioral transition every frame, under the held intent $\bar{\mathbf{s}}_t^h$. Neither transition sees its own observation, so both are predictive rather than reactive. The inference model supplies each posterior with the missing evidence,

$$q_\phi(S \mid \mathbf{o}, \mathbf{c}, \mathbf{a}) = \prod_{t \in \mathcal{T}_c} q_\phi(\mathbf{s}_t^h \mid \mathbf{s}_{t-K}^h, \mathbf{u}_{t-K}, \mathbf{s}_{t-K:t-1}^l, \mathbf{c}_t) \prod_{t=0}^{T} q_\phi(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h, \mathbf{o}_t). \quad (2)$$

For notational convenience, the first factor of each product is taken as the initial distribution: $p_\theta(\mathbf{s}_0^h)$ and $p_\theta(\mathbf{s}_0^l \mid \mathbf{s}_0^h)$, rather than the general recursive form, and likewise for q_ϕ. The posterior transitions differ from the priors only in their additional evidence: $\mathbf{c}_t$ above and $\mathbf{o}_t$ below.

This coupling proceeds along two paths. The top-down path is the held intent $\bar{\mathbf{s}}_t^h$ that conditions the behavioral transition in Eq. 1: it allows maneuver-level intent to guide BEV transition modeling, future prediction, and planning across multiple control steps without re-querying the VLM. The bottom-up path closes the loop in the reverse direction: the behavioral states $\mathbf{s}_{t-K:t-1}^l$ and action chunk $\mathbf{u}_{t-K}$ jointly condition the cognitive transition in Eq. 1, grounding each cognitive update in scene dynamics and executed motion. DualWM thus amortizes VLM reasoning over K frames while keeping intent aligned with closed-loop behavior.

3.3 Cognition-Behavior Coupled Planner

Cognition and behavior converge in the planner through the unified latent state $\mathbf{s}_t = [\bar{\mathbf{s}}_t^h; \mathbf{s}_t^l]$, which shapes planning from both directions. The cognitive layer supplies the maneuver-level intent every K frames, yet the hallucination and uncertainty inherent to large models leave it unreliable on its own. We therefore anchor intent to physical evidence through the bottom-up update in Eq. 1: local geometry, surrounding agents, and short-term scene evolution inform the cognitive state from which intent probabilities are decoded. This allows scene-consistent intents to be reinforced and conflicting ones to be discounted, while the planner uses the current behavioral state to adapt trajectories rather than rigidly follow an infeasible intent.

In addition, different driving situations place different behavioral demands on the planner. In routine scenes, a smooth base motion prior is sufficient for stable progress. The planner then predicts a base trajectory $\boldsymbol{\tau}_{\text{base},t} \in \mathbb{R}^{6\times 2}$. By contrast, interaction-heavy or safety-critical scenes require behavior-specific adjustments: braking, bypassing, merging, and turning, which naturally align with the interactive skill taxonomy of Bench2Drive (Jia et al., 2024). We therefore augment the base trajectory with an intent-gated residual expert for each behavior. The experts predict residual corrections $\boldsymbol{\Delta}_{m,t} \in \mathbb{R}^{6\times 2}$ from the joint state $\mathbf{s}_t$, while a router maps the refined intent to expert weights $g_{m,t} \in [0, 1]$. The final trajectory compensates the base with the weighted residuals,

$$\boldsymbol{\tau}_t = \boldsymbol{\tau}_{\text{base},t} + \sum_{m=1}^{4} g_{m,t} \boldsymbol{\Delta}_{m,t}. \quad (3)$$

Overall, the behavioral state constrains cognitive intent with physical evidence, suppressing the hallucinations to which VLMs are prone, while the intent-gated experts translate the grounded intent into trajectory compensation.

3.4 Training Mechanism

Variational objective. The model is trained by variational inference: the marginal likelihood is intractable, so we maximize the evidence lower bound (ELBO) obtained with the variational posterior q_ϕ of Eq. 2. Below, $\cdot$ denotes the transition arguments shared by p_θ and q_ϕ in Eqs. 1–2, so that

only the extra evidence of each posterior is written out,

$$
\begin{aligned}
&\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c} \mid \mathbf{a}) = \log \int p_\theta(X, S \mid \mathbf{a})\, \mathrm{d}\mathbf{s}^h_{\mathcal{T}_c}\, \mathrm{d}\mathbf{s}^l_{0:T} \\
&= \log \mathbb{E}_{S\sim q_\phi}\Big[\tfrac{p_\theta(X,S\mid\mathbf{a})}{q_\phi(S\mid\mathbf{o},\mathbf{c},\mathbf{a})}\Big] \\
&\geq \mathbb{E}_{S\sim q_\phi}\big[\log p_\theta(X, S \mid \mathbf{a}) - \log q_\phi(S \mid \mathbf{o}, \mathbf{c}, \mathbf{a})\big] \\
&= \mathbb{E}_{S\sim q_\phi}\Big[\sum_{t\in\mathcal{T}_c}\Big(\log p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) + \log \tfrac{p_\theta(\mathbf{s}^h_t\mid\cdot)}{q_\phi(\mathbf{s}^h_t\mid\cdot,\mathbf{c}_t)}\Big) + \sum_{t=0}^{T}\Big(\log p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t) + \log \tfrac{p_\theta(\mathbf{s}^l_t\mid\cdot)}{q_\phi(\mathbf{s}^l_t\mid\cdot,\mathbf{o}_t)}\Big)\Big] \\
&= \mathbb{E}_{S\sim q_\phi}\Big[\underbrace{\textstyle\sum_{t\in\mathcal{T}_c}\Big(\log p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) - \mathcal{D}^h_t\Big)}_{\text{cognitive, every } K \text{ frames}} + \underbrace{\textstyle\sum_{t=0}^{T}\Big(\log p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t) - \mathcal{D}^l_t\Big)}_{\text{behavioral, every frame}}\Big] =: \mathcal{J}_{\mathrm{ELBO}},
\end{aligned}
\tag{4}
$$

where each divergence measures this extra evidence:

$$
\mathcal{D}^h_t = D_{\mathrm{KL}}\big(q_\phi(\mathbf{s}^h_t \mid \cdot, \mathbf{c}_t)\,\|\,p_\theta(\mathbf{s}^h_t \mid \cdot)\big), \qquad \mathcal{D}^l_t = D_{\mathrm{KL}}\big(q_\phi(\mathbf{s}^l_t \mid \cdot, \mathbf{o}_t)\,\|\,p_\theta(\mathbf{s}^l_t \mid \cdot)\big). \tag{5}
$$

The two sums inherit the temporal scales of Eq. 1.

These divergences couple the layers during learning. The behavioral divergence forces the intent-conditioned prior to anticipate the BEV observation, so intent is useful only insofar as it predicts how the scene will actually evolve. The cognitive divergence regularizes the VLM-conditioned posterior against a prior informed by behavioral states and executed actions, so the behavioral layer constrains the uncertainty of VLM-derived cognition. Instantiating the emissions with their decoders turns the two likelihood terms into a BEV reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ and an intent classification loss $\mathcal{L}_{\mathrm{int}}$, and we write $\mathcal{L}_{\mathrm{KL}}$ for the accumulated divergences. Since planning requires anticipation rather than reconstruction alone, we further evaluate the same likelihood on states rolled forward by the behavioral prior, giving a future-prediction term $\mathcal{L}_{\mathrm{pred}}$.

Planning objective. The MoE planner is trained by imitation. With $\boldsymbol{\tau}_{t,k}$ the k-th waypoint at frame t, $\boldsymbol{\tau}^\star_{t,k}$ its expert counterpart, and $\mathcal{M}$ the valid pairs (t,k), the planning loss

$$
\mathcal{L}_{\mathrm{plan}} = \frac{1}{|\mathcal{M}|}\sum_{(t,k)\in\mathcal{M}} \big\|\boldsymbol{\tau}_{t,k} - \boldsymbol{\tau}^\star_{t,k}\big\|^2 \tag{6}
$$

supervises both trajectories, with the base trajectory detached in the expert-gated output. A gate loss $\mathcal{L}_{\mathrm{gate}}$ further aligns the expert weights $g_{m,t}$ with the annotated maneuver, so that each expert responds only to its own behavior. The full objective is

$$
\mathcal{L} = -\,\mathcal{J}_{\mathrm{ELBO}} + \lambda_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{plan}}\mathcal{L}_{\mathrm{plan}} + \lambda_{\mathrm{gate}}\mathcal{L}_{\mathrm{gate}} + \mathcal{L}_{\mathrm{aux}}, \tag{7}
$$

where $-\mathcal{J}_{\mathrm{ELBO}}$ expands into $\mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{int}} + \mathcal{L}_{\mathrm{KL}}$, and $\mathcal{L}_{\mathrm{aux}}$ collects the auxiliary regularizers. All stochastic states are sampled with the reparameterization trick, the expectations are estimated with a single posterior sample, and the VLM and BEV encoders remain frozen throughout. In practice, we optimize Eq. 7 in two stages. The cognitive layer and its intent heads are trained first. Both layers are then jointly optimized with the behavioral decoders and MoE planner under the remaining loss terms, while the intent heads remain frozen. More details are provided in the Appendix.

Asynchronous inference. At deployment, DualWM runs in closed loop. The VLM is queried only on $\mathcal{T}_c$; between queries the most recent cognitive state continues to condition the behavioral layer. At every frame, the behavioral posterior updates $\mathbf{s}^l_t$ from the BEV observation $\mathbf{o}_t$ and the previous action, the MoE planner generates the trajectory $\boldsymbol{\tau}_t$ from the joint state $\mathbf{s}_t$, and a controller tracks it to emit the low-level action. The actions executed over one interval form the chunk $\mathbf{u}_t$ that drives the cognitive update at frame $t+K$.

4 Experiments

4.1 Experimental Setup

Dataset and metrics. We train and evaluate DualWM on Bench2Drive (Jia et al., 2024), a closed-loop benchmark for end-to-end autonomous driving in CARLA v2. For fair comparison with all

Table 1: Closed-loop results of end-to-end autonomous driving methods on Bench2Drive under the official *base* training set. C/L refers to camera/LiDAR. * denotes expert feature distillation.

Method	Reference	Scheme	Modality	Closed-loop Metric ↑			
				DS	**SR** (%)	Efficiency	Comfortness
TCP*	NeurIPS 22	IL	C	40.70	15.00	54.26	**47.80**
VAD	ICCV 23	IL	C	42.35	15.00	157.94	46.01
UniAD-Base	CVPR 23	IL	C	45.81	16.36	129.21	43.58
TCP-traj*	NeurIPS 22	IL	C	59.90	30.00	76.54	18.08
ThinkTwice*	CVPR 23	IL	C	62.44	31.23	69.33	16.22
DriveTransformer	ICLR 25	IL	C	63.46	35.01	100.64	20.78
DriveAdapter*	ICCV 23	IL	C&L	64.22	33.08	70.22	16.01
Raw2Drive	NeurIPS 25	RL	C	71.36	50.24	**214.17**	22.42
ReCogDrive	ICLR 26	RL	C	71.36	45.45	138.18	17.45
DriveMoE	CVPR 26	IL	C	74.22	48.64	175.96	15.31
ORION	ICCV 25	IL	C	77.74	54.62	151.48	17.38
MindDrive	ECCV 26	RL	C	78.04	55.09	–	–
LVDrive	arXiv 26	IL	C	80.71	58.26	155.77	14.34
DualWM (Ours)	–	IL	C	**81.22**	**61.51**	177.10	22.05

baselines, we use the official *base* training set, which contains 1000 clips, each covering approximately 150 m of continuous driving in one traffic scene. We split these clips into 950 for training and 50 for validation. Closed-loop evaluation is conducted on the official 220 short routes, covering 44 interactive scenarios with 5 routes per scenario. We report the five official metrics: Driving Score (DS), which combines route completion with infraction penalties; Success Rate (SR), the fraction of routes completed within the time limit; Efficiency; Comfortness; and Multi-Ability, a per-skill breakdown over five advanced urban driving abilities.

Implementation settings. Following the Bench2Drive baselines, DualWM takes only surround-view cameras as input. We employ a frozen BEVFormer (Li et al., 2022) as the observation encoder, adopting the official Bench2Drive checkpoint, and a frozen ORION (Fu et al., 2025) as the cognitive backbone, which is queried every $K = 10$ frames. We follow the two-stage schedule with a batch size of 16 and a learning rate of 1×10^{-4}. All main experiments run on a single NVIDIA A800-80GB GPU and are repeated with three random seeds. More details are provided in the Appendix.

4.2 Main Results

As reported in Table 1, DualWM achieves the highest driving score on Bench2Drive, reaching 81.22 DS and 61.51% SR averaged over three random seeds. It outperforms recent VLM-based methods such as LVDrive, MindDrive, and ORION in both driving score and success rate, and its improvement over classical end-to-end methods is even larger, exceeding DriveTransformer by 17.76 DS. Notably, DualWM reuses ORION as its frozen cognitive backbone, querying it only once every ten frames, yet surpasses ORION's own driving score by 3.48 DS and success rate by 6.89 points. These results show that the asynchronous, world-model design of DualWM does not compromise closed-loop accuracy: a slowly varying cognitive intent coupled with a grounded behavioral model already matches the best driving performance on this benchmark.

Additionally, Table 2 reports the multi-ability results. DualWM attains the highest mean ability of 61.82%, exceeding ORION and MindDrive by 7.10 and 4.88 percentage points, respectively. Specifically, DualWM ranks first in Merging (55.00%) and Give Way (56.67%), which require deciding when and how to change or yield lanes in interactive traffic. This is where direct VLM planners fall short: ORION reaches only 25.00% and 30.00% on these two abilities, because the timing of a lane change is diverse and difficult to map from reasoning to a trajectory in a single step. This advantage supports the role of the intent-gated MoE planner: the cognitive intent selects the relevant behavior expert, and the grounded behavioral state governs its execution, converting an ambiguous maneuver into a structured decision. In contrast, DualWM trails ORION and LVDrive on Traffic Sign

Table 2: Multi-ability success rates on Bench2Drive under the official base training set.

Method	Scheme	Ability (%) ↑					
		Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	Mean
TCP*	IL	16.18	20.00	20.00	10.00	6.99	14.63
VAD	IL	8.11	24.44	18.64	20.00	19.15	18.07
UniAD-Base	IL	14.10	17.78	21.67	10.00	14.21	15.55
TCP-traj*	IL	8.89	24.29	51.67	40.00	46.28	34.22
ThinkTwice*	IL	27.38	18.42	35.82	50.00	54.23	37.17
DriveTransformer	IL	17.57	35.00	48.36	40.00	52.10	38.60
DriveAdapter*	IL	28.82	26.38	48.76	50.00	56.43	42.08
Raw2Drive	RL	43.35	51.11	60.00	50.00	62.26	53.34
ReCogDrive	RL	29.73	20.00	69.09	20.00	71.34	42.03
DriveMoE	IL	34.67	40.00	65.45	40.00	59.44	47.91
ORION	IL	25.00	71.11	**78.33**	30.00	69.15	54.72
MindDrive	RL	32.89	**75.56**	68.33	50.00	57.89	56.94
LVDrive	IL	39.74	68.89	71.67	20.00	**74.21**	54.90
DualWM (Ours)	IL	**55.00**	71.11	66.67	**56.67**	59.65	**61.82**

Table 3: Inference latency on Bench2Drive.

Method	DS ↑	VLM	Latency (ms)↓
ThinkTwice*	62.44	×	762
DriveAdapter*	64.22	×	931
DriveMoE†	74.22	✓	1116
ORION†	77.74	✓	1493
MindDrive†	78.04	✓	1091
LVDrive	80.71	✓	2030
DualWM (Ours)†	81.22	✓	551

Table 4: Ablation on Bench2Drive. Δ_{DS} is the DS change from the full model.

Variant	DS↑	Δ_{DS}	SR(%)↑	Ability↑
DualWM (full)	**81.22**	–	**61.51**	**61.82**
w/o cognition	62.88	−18.34	33.64	33.21
w/o feedback	76.05	−5.17	48.18	45.08
w/o prediction	69.55	−11.67	42.73	40.60
w/o MoE	72.45	−8.77	49.09	48.01

(59.65%) and Emergency Brake (66.67%). These abilities require immediate perceptual responses to signals and sudden hazards, but low-frequency cognitive updates may cause slight delays.

More importantly, DualWM attains this driving quality at a markedly lower inference cost. Table 3 compares inference latency on Bench2Drive. VLM marks methods using a vision-language backbone during inference, and † denotes VLM-based methods with publicly released models evaluated on the same hardware as DualWM. Among the compared VLM-based methods, DualWM has the lowest latency at 551 ms per step, roughly 36.9% of that of ORION (1493 ms) and 27.1% of that of LVDrive (2030 ms). Its latency is even approximately 27.7% lower than that of the fastest non-VLM baseline, ThinkTwice (762 ms). This efficiency stems from the asynchronous design: the VLM is queried every ten frames, the most recent cognitive state is reused throughout each query interval, and the behavioral layer and MoE planner operate every step. Together with Table 1, these results demonstrate state-of-the-art driving quality at substantially lower measured inference latency, making high-level cognition considerably more practical for closed-loop driving.

4.3 Ablation Studies

Table 4 isolates the contribution of each component by removing it from DualWM. The cognitive layer has the largest impact: removing it lowers DS by 18.34 and SR to 33.64%. Here the behavioral layer still observes the BEV scene but no longer receives persistent semantic guidance, so it reacts to the immediate surroundings without a maneuver-level goal. Without bottom-up feedback, cognitive inference no longer receives timely feedback on behavioral states, and DS decreases by 5.17 points. Disabling predictive modeling in both layers reduces DS by 11.67, highlighting the value of the dual-



Figure 4: Qualitative results of DualWM. (a) A left turn at a stop-sign junction interrupted midway by a crossing pedestrian: high-level intents guide the low-level plan from yielding to completing the turn. (b) A queue blocking the ego lane, passable only through the oncoming lane: the low-level layer declines an infeasible high-level intent and plans the bypass on its own.

layer world model's predictive representations for closed-loop planning. Removing the MoE planner lowers DS by 8.77, as a single shared policy must cover all behaviors and cannot specialize to demanding maneuvers such as merging or bypassing. These ablations support complementary roles for the four mechanisms: the cognitive layer supplies maneuver-level intent, bottom-up feedback conditions cognitive updates on behavioral states, prediction grounds planning in anticipated scene dynamics, and the MoE planner maps the joint state to behavior-specific control.

4.4 Qualitative Results

Figure 4 demonstrates that the proposed method effectively constrains hallucinated outputs from the VLM-based cognitive layer using low-level behavioral features, leading to safer and more reasonable driving behaviors. Figure 4(a) illustrates the case where the high-level cognitive outputs from the upper layer align with the real-time predictions from the lower layer. In such a scenario, the upper network guides the lower behavioral module to generate a continuous sequence of navigation actions. These include decelerating at a stop sign, initiating a turn, yielding to crossing pedestrians, and resuming the turn. In this way, the behavioral fluctuations that might otherwise arise from variations in the lower network's real-time judgments are avoided. Figure 4(b) illustrates a case where the high-level cognitive outputs from the upper layer conflict with the real-time predictions from the lower layer. In response, the lower behavioral module corrects the infeasible intent from the upper layer: instead of executing the suggested right lane change, the vehicle performs an overtaking maneuver through the oncoming lane. Overall, the proposed method capitalizes on VLM-based reasoning at the cognitive intent layer to provide consistent long-term guidance to the policy network, while the lower behavioral dynamics layer imposes real-time constraints on hallucinated VLM outputs, thereby improving driving safety and reliability.

5 Conclusion

In this paper, we propose DualWM, an end-to-end autonomous driving framework centered on an asynchronous dual-layer latent world model. Within a unified latent space, the model bidirectionally couples slow VLM cognition with fast behavioral dynamics, enabling intent-guided behavioral prediction while grounded dynamics constrain VLM-derived cognitive uncertainty. A cognition-behavior coupled planner augments a base trajectory with intent-gated expert residuals to generate the ego trajectory. On Bench2Drive, DualWM achieves strong closed-loop performance with substantially lower inference latency than VLM-based baselines. Future work will explore adapting the VLM query interval to scene urgency for a more timely response in safety-critical situations, and compressing and distilling the VLM backbone to bring inference latency closer to real-time.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript preparation, including grammar checking, improving sentence clarity, making the writing more concise, and refining transitions between sentences and paragraphs. They also provided suggestions on figure layout and visual presentation, and assisted with organizing experimental results and checking numerical calculations. The authors retained control over the final content and presentation and take responsibility for the manuscript, including all AI-assisted content.

ETHICS STATEMENT

This work presents DualWM, a latent world model that couples cognitive intent with behavioral dynamics for autonomous driving. The pretrained vision-language model supports driving reasoning and planning, and our experiments are conducted on Bench2Drive in the CARLA simulator without deployment on public roads. Real-world deployment would require further safety validation and appropriate safeguards. In summary, we believe that the simulation-based study presented in this paper does not raise any ethical concerns.

REPRODUCIBILITY STATEMENT

To support reproducibility, Appendix A presents the probabilistic formulation and derivations, and Appendix B describes the network architectures. Appendix C documents the training and inference procedures, data split, evaluation metrics, hyperparameters, and computational infrastructure. Appendix D provides per-seed evaluation results and additional analyses, while Appendix E presents qualitative reconstructions and predictions. We provide partial code for the core components in the supplementary material. The complete codebase will be open-sourced upon acceptance.

REFERENCES

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a. doi: 10.1109/TPAMI.2024.3435937.

Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. Asynchronous large language model enhanced planner for autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 22–38, 2024b.

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 *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700, 2018.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. ORION: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834, 2025.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Dingkang Liang, and Xiang Bai. MindDrive: A vision-language-action model for autonomous driving via online reinforcement learning. In *European Conference on Computer Vision (ECCV)*, 2026.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 91560–91596, 2024. doi: 10.52202/079017-2906.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Shadi Hamdan, Chonghao Sima, Zetong Yang, Hongyang Li, and Fatma Güney. ETA: Efficiency through thinking ahead, a dual approach to self-driving with large models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26529–26538, 2025.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 20703–20716, 2022a. doi: 10.52202/068431-1505.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving, 2023a.

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 (ECCV)*, pp. 533–549, 2022b. doi: 10.1007/978-3-031-19839-7_31.

Tianshuai Hu, Xiaolu Liu, Song Wang, Yiyao Zhu, Ao Liang, Lingdong Kong, Guoyang Zhao, Zeying Gong, Jun Cen, Zhiyu Huang, Xiaoshuai Hao, Linfeng Li, Hang Song, Xiangtai Li, Jun Ma, Shaojie Shen, Jianke Zhu, Dacheng Tao, Ziwei Liu, and Junwei Liang. Vision-language-action models for autonomous driving: Past, present, and future, 2025.

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 (CVPR)*, pp. 17853–17862, 2023b.

Zilin Huang, Zihao Sheng, Zhengyang Wan, Yansong Qu, Junwei You, Sicong Jiang, and Sikai Chen. DriveVLM-RL: Neuroscience-inspired reinforcement learning with vision-language models for safe and deployable autonomous driving, 2026.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. 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)*, pp. 7953–7963, 2023a.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21983–21994, 2023b.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 819–844, 2024. doi: 10.52202/079017-0025.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. In *International Conference on Learning Representations (ICLR)*, 2026.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6114.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters, 2015. URL https://arxiv.org/abs/1511.05121.

Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/08058bf500242562c0d031ff830ad094-Abstract.html.

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 *International Conference on Learning Representations (ICLR)*, 2025a.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146, 2025b.

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. In *International Conference on Learning Representations (ICLR)*, 2026.

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.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, volume 13669 of *Lecture Notes in Computer Science*, pp. 1–18. Springer, 2022. doi: 10.1007/978-3-031-20077-9_1.

Wenru Liu, Pei Liu, and Jun Ma. DSDrive: Distilling large language model for lightweight end-to-end autonomous driving with unified reasoning and planning, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Xiaodong Mei, Diankun Zhang, Hongwei Xie, Guang Chen, Hangjun Ye, and Dan Xu. LVDrive: Latent visual representation enhanced vision-language-action autonomous driving model, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15522–15533, 2024.

Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *European Conference on Computer Vision (ECCV)*, volume 15110 of *Lecture Notes in Computer Science*, pp. 256–274. Springer, 2024. doi: 10.1007/978-3-031-72943-0_15.

Shuhan Tan, Kashyap Chitta, Yuxiao Chen, Ran Tian, Yurong You, Yan Wang, Wenjie Luo, Yulong Cao, Philipp Krähenbühl, Marco Pavone, and Boris Ivanovic. Latent chain-of-thought world modeling for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39724–39733, 2026.

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 (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726, 2025.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision (ECCV)*, volume 15106 of *Lecture Notes in Computer Science*, pp. 55–72. Springer, 2024a. doi: 10.1007/978-3-031-73195-2_4.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024b.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. UniDrive-WM: Unified understanding, planning and generation world model for autonomous driving. In *Computer Vision – ECCV 2026*, volume 17017 of *Lecture Notes in Computer Science*, pp. 373–391. Springer, 2026. doi: 10.1007/978-3-032-37252-9_21.

Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2Drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in CARLA v2). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. DriveMoE: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10678–10688, 2026.

Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Copilot4D: Learning unsupervised world models for autonomous driving via discrete diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Ruifei Zhang, Junlin Xie, Wei Zhang, Weikai Chen, Xiao Tan, Xiang Wan, and Guanbin Li. AdaDrive: Self-adaptive slow-fast system for language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5112–5121, 2025.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3d occupancy world model for autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 55–72, 2024a.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. GenAD: Generative end-to-end autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 87–104, 2024b.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, 2025.

OVERVIEW

The appendix is organized as follows:

- Section A: Derivation of the evidence lower bound and training objectives.
- Section B: Network architectures.
- Section C: Training and evaluation details.
- Section D: Ablation studies, query-interval sensitivity, and per-seed results.
- Section E: Qualitative reconstructions and predictions.

A DERIVATION OF THE EVIDENCE LOWER BOUND AND TRAINING OBJECTIVES

This appendix derives the evidence lower bound of the model in full, states the emission distributions of its two layers, and records every point at which the implemented training objective departs from the bound. The bound gives the two-layer world model a single principled training objective over its representation-learning terms: reconstruction, maneuver supervision, and the latent divergences. The planner is trained by discriminative losses on the representation this objective learns, and every such term is accounted for below.

A.1 ASSUMPTIONS AND NOTATION

Control frames are indexed by $t = 0, \dots, T$. The VLM is queried on $\mathcal{T}_c = \{0, K, 2K, \dots\}$ with $K = 10$, one second of driving at the 10 Hz rate of the dataset, and $\kappa(t) = K\lfloor t/K \rfloor$ is the most recent query frame.

We treat the executed action sequence $\mathbf{a} = (\mathbf{a}_0, \dots, \mathbf{a}_{T-1})$ as an exogenous conditioning variable rather than a modeled random variable, so the model specifies a conditional distribution given $\mathbf{a}$ and never a distribution over $\mathbf{a}$ itself. During training, $\mathbf{a}$ is the recorded expert action sequence. Each chunk $\mathbf{u}_t = (\mathbf{a}_t, \dots, \mathbf{a}_{t+K-1})$ is a deterministic function of $\mathbf{a}$, so conditioning on $\mathbf{a}$ already subsumes conditioning on every $\mathbf{u}$. The held cognitive state $\bar{\mathbf{s}}^h_t = \mathbf{s}^h_{\kappa(t)}$ is likewise a deterministic copy of an existing latent variable rather than a variable of its own, so the model contains exactly $|\mathcal{T}_c| + T + 1$ stochastic states and the marginalization below integrates over these.

For the behavioral layer, $\mathbf{o}_t$ denotes the encoded sensor observation when conditioning the posterior and the decoder targets specified in Eq. (18) when evaluating the emission likelihood. Decoder targets, including the expert future-path channel, are used only for supervision. The variable $\mathbf{y}_t$ denotes the maneuver annotation available on a query frame. The annotation is a supervision label rather than a sensor reading, which places the cognitive layer in the standard semi-supervised setting: the label is an emission of the cognitive state, and that state is inferred from the VLM feature $\mathbf{c}_t$. The derivation follows the standard variational recipe for sequential latent-variable models (Kingma & Welling, 2014; Krishnan et al., 2015).

A.2 MODEL FACTORIZATION

Written with the initial distributions made explicit, the generative model is

$$\begin{aligned} p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{a}) &= p_\theta(\mathbf{s}^h_0)\, p_\theta(\mathbf{y}_0 \mid \mathbf{s}^h_0)\, p_\theta(\mathbf{s}^l_0 \mid \mathbf{s}^h_0)\, p_\theta(\mathbf{o}_0 \mid \mathbf{s}^l_0) \\ &\quad \times \textstyle\prod_{t \in \mathcal{T}_c \setminus \{0\}} p_\theta(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1})\, p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) \\ &\quad \times \textstyle\prod_{t=1}^{T} p_\theta(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t)\, p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t), \end{aligned} \tag{8}$$

and the variational distribution is

$$\begin{aligned} q_\phi(\mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{o}_{0:T}, \mathbf{c}_{\mathcal{T}_c}, \mathbf{a}) &= q_\phi(\mathbf{s}^h_0 \mid \mathbf{c}_0)\, q_\phi(\mathbf{s}^l_0 \mid \mathbf{s}^h_0, \mathbf{o}_0) \\ &\quad \times \textstyle\prod_{t \in \mathcal{T}_c \setminus \{0\}} q_\phi(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1}, \mathbf{c}_t) \times \textstyle\prod_{t=1}^{T} q_\phi(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t, \mathbf{o}_t). \end{aligned} \tag{9}$$

Each posterior factor carries exactly one argument that its prior counterpart lacks, which is $\mathbf{c}_t$ in the upper factor and $\mathbf{o}_t$ in the lower. The variational distribution q_ϕ carries no emission factors because

the emissions are not latent. Neither transition in (8) is conditioned on its own emission target, so both are predictive rather than reactive.

A.3 Derivation of the Bound

We adopt the convention that at $t = 0$ the cognitive transition denotes $p_\theta(\mathbf{s}_0^h)$ and the behavioral transition $p_\theta(\mathbf{s}_0^l \mid \mathbf{s}_0^h)$, and likewise for q_ϕ, so that both models are single products over $\mathcal{T}_c$ and over $0{:}T$. Marginalizing the latent states and reweighting by q_ϕ,

$$\begin{aligned}
&\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c} \mid \mathbf{a}) \\
&= \log \int p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{a}) \, \mathrm{d}\mathbf{s}^h_{\mathcal{T}_c} \, \mathrm{d}\mathbf{s}^l_{0:T} \\
&= \log \mathbb{E}_{q_\phi}\left[\frac{p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{a})}{q_\phi(\mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{o}_{0:T}, \mathbf{c}_{\mathcal{T}_c}, \mathbf{a})} \right] \qquad (10) \\
&\geq \mathbb{E}_{q_\phi}\big[\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{a}) \\
&\qquad\qquad - \log q_\phi(\mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{o}_{0:T}, \mathbf{c}_{\mathcal{T}_c}, \mathbf{a}) \big], \qquad (11)
\end{aligned}$$

the reweighting is valid wherever q_ϕ dominates p_θ, and the last step is Jensen's inequality. Nothing about the two timescales has been used so far, and (11) is the generic variational bound (Kingma & Welling, 2014). Taking logarithms of (8) and (9) converts both products into sums,

$$\begin{aligned}
&\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{a}) \\
&\quad = \textstyle\sum_{t \in \mathcal{T}_c} \big[\log p_\theta(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1}) + \log p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) \big] \\
&\quad + \textstyle\sum_{t=0}^{T} \big[\log p_\theta(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t) + \log p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t) \big], \qquad (12) \\
&\log q_\phi(\mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{o}_{0:T}, \mathbf{c}_{\mathcal{T}_c}, \mathbf{a}) = \textstyle\sum_{t \in \mathcal{T}_c} \log q_\phi(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1}, \mathbf{c}_t) \\
&\quad + \textstyle\sum_{t=0}^{T} \log q_\phi(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t, \mathbf{o}_t).
\end{aligned}$$

The transition sums of p_θ run over the same two index sets as those of q_ϕ, so subtracting them pairs each prior transition with the posterior at the same frame, while the emission terms of p_θ have no counterpart to cancel against. Regrouping by frame,

$$\begin{aligned}
&\mathbb{E}_{q_\phi}\Big[\textstyle\sum_{t \in \mathcal{T}_c} \Big(\log p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) + \log \frac{p_\theta(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1})}{q_\phi(\mathbf{s}^h_t \mid \mathbf{s}^h_{t-K}, \mathbf{u}_{t-K}, \mathbf{s}^l_{t-K:t-1}, \mathbf{c}_t)} \Big) \\
&\quad + \textstyle\sum_{t=0}^{T} \Big(\log p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t) + \log \frac{p_\theta(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t)}{q_\phi(\mathbf{s}^l_t \mid \mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t, \mathbf{o}_t)} \Big) \Big] \qquad (13) \\
&= \mathbb{E}_{q_\phi}\Big[\textstyle\sum_{t \in \mathcal{T}_c} \big(\log p_\theta(\mathbf{y}_t \mid \mathbf{s}^h_t) - \mathcal{D}^h_t \big) + \sum_{t=0}^{T} \big(\log p_\theta(\mathbf{o}_t \mid \mathbf{s}^l_t) - \mathcal{D}^l_t \big) \Big] = \mathcal{J}_{\mathrm{ELBO}}, \qquad (14)
\end{aligned}$$

where $\mathcal{D}^h_t$ and $\mathcal{D}^l_t$ are the Kullback-Leibler divergences of the cognitive and behavioral transitions, read at $t = 0$ with the initial distributions in place of the transitions. Rewriting each expected log-ratio as $-\mathcal{D}_t$ uses the tower property established below. The asynchrony enters at (13), where the two index sets separate a cognitive term charged once per query interval from a behavioral term charged once per frame. No term cancels between the two models, since each posterior differs from its prior by exactly one extra argument, so every transition contributes a divergence. Structurally, (14) is the sequential ELBO of latent world models (Lee et al., 2020; Hafner et al., 2020; Hu et al., 2022a), extended here to a two-timescale latent process. A slow cognitive layer driven by a vision-language model and a fast behavioral layer driven by observations are cast as one latent-variable world model whose asynchronous training reduces to a single principled objective. Casting the two layers this way is what lets one bound account for reconstruction and divergence at the two rates, and it is the basis on which the cognitive and behavioral layers are coupled within a single objective rather than treated in isolation.

The step from (13) to (14) relies on the autoregressive structure of q_ϕ. The conditional law of $\mathbf{s}^l_t$ given $(\mathbf{s}^l_{t-1}, \mathbf{a}_{t-1}, \bar{\mathbf{s}}^h_t)$ does not depend on later states, so the expectation can be taken in two stages, inner over $\mathbf{s}^l_t$ with its conditioning variables held fixed and outer over those variables under q_ϕ. By

the tower property,

$$\begin{aligned}&\mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h)}{q_\phi(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h, \mathbf{o}_t)}\right] \\ &\quad = \mathbb{E}_{q_\phi}\left[\mathbb{E}_{\mathbf{s}_t^l \sim q_\phi}\left[\log \frac{p_\theta(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h)}{q_\phi(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h, \mathbf{o}_t)}\right]\right] = -\,\mathbb{E}_{q_\phi}\big[\mathcal{D}_t^l\big],\end{aligned} \tag{15}$$

the inner expectation being a negative Kullback-Leibler divergence by definition. The divergence stays inside the outer expectation because it is a function of the conditioning variables, which are themselves random under q_ϕ. The cognitive ratio is handled the same way with $\mathbf{c}_t$ in place of $\mathbf{o}_t$. No divergence between the two layers appears anywhere in (14): the layers are coupled through the conditioning arguments of their transitions, not through an additional penalty term.

The slack in the bound takes the familiar form. Subtracting (13) from $\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c} \mid \mathbf{a})$ and applying the chain rule of probability to the joint density,

$$\log p_\theta(\mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c} \mid \mathbf{a}) - \mathcal{J}_{\mathrm{ELBO}} = D_{\mathrm{KL}}\big(q_\phi \,\big\|\, p_\theta(\mathbf{s}^h_{\mathcal{T}_c}, \mathbf{s}^l_{0:T} \mid \mathbf{o}_{0:T}, \mathbf{y}_{\mathcal{T}_c}, \mathbf{a})\big) \;\geq\; 0, \tag{16}$$

which re-derives the bound without Jensen and demonstrates that it is tight exactly when the variational posterior matches the true posterior almost everywhere. The bound holds for every query interval K. At $K = 1$ the held state reduces to $\bar{\mathbf{s}}_t^h = \mathbf{s}_t^h$ and the two index sets coincide, leaving a synchronous two-layer bound rather than a single-layer one. For $K > 1$ the cognitive sum thins while the behavioral does not, so K alters the generative model, not the bound's validity. The selection of K is described in Appendix C.4.

Maximizing (14) drives each prior to predict the evidence its posterior receives, which is the coupling between the two layers. The bound makes one asymmetry of that coupling explicit. The top-down path runs through the conditioning argument $\bar{\mathbf{s}}_t^h$ of the behavioral transition, which is a forward dependency present in every sampled trajectory. The bottom-up path runs through the behavioral states $\mathbf{s}^l_{t-K:t-1}$ and action chunk $\mathbf{u}_{t-K}$, which jointly condition the cognitive transition on scene dynamics and executed motion. Gradients propagate through both paths during joint training.

A.4 Emission Distributions

The model writes the two emissions as single factors. Both are products of conditionally independent heads, which is why the corresponding losses are sums.

The cognitive emission decomposes into two categorical variables and six independent binary ones,

$$p_\theta(\mathbf{y}_t \mid \mathbf{s}_t^h) = p_\theta(y_t^{\mathrm{lat}} \mid \mathbf{s}_t^h)\, p_\theta(y_t^{\mathrm{lon}} \mid \mathbf{s}_t^h) \times \textstyle\prod_{j=1}^{6} p_\theta(y_{t,j}^{\mathrm{rec}} \mid \mathbf{s}_t^h), \tag{17}$$

where y_t^{lat} ranges over five lateral maneuvers (left turn, left lane change, straight, right turn, right lane change), y_t^{lon} over four longitudinal ones (stopped, accelerating, decelerating, keeping speed), and the six binary variables $y_{t,j}^{\mathrm{rec}}$ mark recognized situations that pair an observation with the maneuver it calls for: safety brake, sign stop, crossing brake, emergency yield, obstacle bypass, and cyclist overtake. The two categorical heads are softmax, and the six recognition heads are independent sigmoids, so a frame may carry several recognized situations at once. Taking the logarithm of (17) makes $\mathcal{L}_{\mathrm{int}}$ a sum of two multi-class and six binary cross-entropies.

The behavioral emission decomposes over decoder heads on two views. Writing $\mathbf{o}_t = (\mathbf{o}_t^{\mathrm{bev}}, \mathbf{o}_t^{\mathrm{front}})$,

$$p_\theta(\mathbf{o}_t \mid \mathbf{s}_t^l) = \textstyle\prod_{v \in \{\mathrm{bev},\mathrm{front}\}} \prod_{k \in \mathcal{H}_v} p_\theta(\mathbf{o}_t^{v,k} \mid \mathbf{s}_t^l), \tag{18}$$

where $\mathcal{H}_{\mathrm{bev}}$ contains a seven-class semantic layout, an instance center map, an instance offset field, a scene flow field, and the ego future-path channel, and $\mathcal{H}_{\mathrm{front}}$ contains eight-class front-view semantics and depth. Categorical heads are per-cell categorical distributions whose negative log-likelihood is cross-entropy. The center map is Gaussian with fixed variance, and the offset, flow, and depth heads are Laplace with fixed scale, so their negative log-likelihoods reduce to squared and absolute errors up to additive constants. $\mathcal{L}_{\mathrm{rec}}$ is the weighted sum of these terms. BEV targets cover $\pm 40\,\mathrm{m}$ around the vehicle in every direction, so the behavioral state is held responsible for the full surround even though only one camera view is reconstructed. The per-head weights appear in Appendix C.

A.5 Gaussian Parameterization

The stochastic component of each transition in (8) and (9) is a diagonal Gaussian whose mean and standard deviation are produced by a network from the conditioning arguments. State densities refer to these stochastic components, with recurrent states implicit in the conditioning. The cognitive state carries 512 stochastic and 512 recurrent dimensions, and the behavioral state 512 stochastic and 1024 recurrent. The stochastic components, denoted by $(\mathbf{s})_{\text{stoch}}$, are sampled with the reparameterization trick (Kingma & Welling, 2014),

$$(\mathbf{s})_{\text{stoch}} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{19}$$

so the gradient of $\mathcal{J}_{\text{ELBO}}$ is unbiased and has the variance of a one-sample estimator. Because both arguments of each divergence are diagonal Gaussians, $\mathcal{D}_t^h$ and $\mathcal{D}_t^l$ need no sampling and are evaluated in closed form,

$$D_{\text{KL}} = \textstyle\sum_{i=1}^{d} \Big[\log \frac{\sigma_{p,i}}{\sigma_{q,i}} + \frac{\sigma_{q,i}^2 + (\mu_{q,i} - \mu_{p,i})^2}{2\sigma_{p,i}^2} - \frac{1}{2} \Big], \tag{20}$$

with (μ_q, σ_q) posterior and (μ_p, σ_p) prior. A floor on σ keeps the logarithm finite under posterior collapse.

A.6 The Prediction Term

The training objective names a future-prediction term $\mathcal{L}_{\text{pred}}$ but leaves its form open. It evaluates the emission (18) on states produced by the behavioral prior rather than the posterior, and it does so at two granularities.

The per-frame term predicts a fixed horizon $H_1 = 10$ frames, one second ahead, from every observed frame. Rolling the prior forward from the posterior state at t,

$$\hat{\mathbf{s}}_{t+h}^l = \mathbb{E}_{p_\theta(\cdot \mid \hat{\mathbf{s}}_{t+h-1}^l, \mathbf{a}_{t+h-1}, \bar{\mathbf{s}}_t^h)}[\mathbf{s}_{t+h}^l], \qquad \hat{\mathbf{s}}_t^l = \mathbf{s}_t^l, \tag{21}$$

the per-frame term scores the one-second-ahead BEV emission from the rolled-out state,

$$\mathcal{L}_{\text{pred}}^{\text{pf}} = -\,\mathbb{E}_{q_\phi} \Big[\textstyle\sum_{t=0}^{T} \log p_\theta \big(\mathbf{o}_{t+H_1}^{\text{bev}} \mid \hat{\mathbf{s}}_{t+H_1}^l \big) \Big]. \tag{22}$$

The multi-horizon term rolls the prior further from the last observed frame, to $H \in \{20, 30\}$ frames, two and three seconds ahead,

$$\mathcal{L}_{\text{pred}}^{\text{mh}} = -\,\mathbb{E}_{q_\phi} \Big[\textstyle\sum_{H \in \{20,30\}} \gamma_H \log p_\theta \big(\mathbf{o}_{T+H}^{\text{bev}} \mid \hat{\mathbf{s}}_{T+H}^l \big) \Big], \tag{23}$$

with γ_H a discount that offsets the growing difficulty of the far horizons, and $\mathcal{L}_{\text{pred}}$ is the weighted sum of the two. Both terms are evaluated on the BEV heads only. A one-step prior spans 0.1 s and is close to trivial, so a horizon of a second or more is required before the latent must represent how other agents move rather than where they currently are. Neither term belongs to $\mathcal{J}_{\text{ELBO}}$: the bound scores the posterior as an encoder, whereas (22) and (23) score the prior as a predictor. The rollout in (21) is conditioned on the recorded expert actions, available during training. The prediction term is a training-time auxiliary and is not evaluated in closed-loop deployment. The prediction term therefore encourages the behavioral state to retain information about how the scene will evolve. At deployment, the planner reads this state directly, so any anticipatory information is already part of the representation it consumes rather than produced by an explicit forward rollout at plan time.

A.7 Departures from the Bound

Writing the three losses $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{int}}$, and $\mathcal{L}_{\text{KL}}$ in terms of (14),

$$\begin{gathered} \mathcal{L}_{\text{rec}} = -\,\mathbb{E}_{q_\phi} \big[\textstyle\sum_{t=0}^{T} \log p_\theta(\mathbf{o}_t \mid \mathbf{s}_t^l) \big], \qquad \mathcal{L}_{\text{int}} = -\,\mathbb{E}_{q_\phi} \big[\textstyle\sum_{t \in \mathcal{T}_c} \log p_\theta(\mathbf{y}_t \mid \mathbf{s}_t^h) \big], \\ \mathcal{L}_{\text{KL}} = \mathbb{E}_{q_\phi} \big[\textstyle\sum_{t \in \mathcal{T}_c} \mathcal{D}_t^h + \sum_{t=0}^{T} \mathcal{D}_t^l \big], \end{gathered} \tag{24}$$

so that $-\mathcal{J}_{\text{ELBO}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{int}} + \mathcal{L}_{\text{KL}}$ exactly. The implemented objective uses stage-specific loss terms and weights, as detailed in Appendix C.

Balanced divergence. Each divergence is implemented as

$$\widetilde{\mathcal{D}} = \alpha\, D_{\mathrm{KL}}\big(\mathrm{sg}[q_\phi] \,\|\, p_\theta\big) + (1-\alpha)\, D_{\mathrm{KL}}\big(q_\phi \,\|\, \mathrm{sg}[p_\theta]\big), \tag{25}$$

with $\alpha = 0.75$, following Hu et al. (2022a), where $\mathrm{sg}[\cdot]$ is the stop-gradient operator. Since sg is the identity in the forward pass, $\widetilde{\mathcal{D}} = \mathcal{D}$ for every α, so the value of the objective is unchanged and the bound continues to hold. Only the gradient is redistributed, a fraction α to the prior and $1-\alpha$ to the posterior, which keeps the prior from being dragged by a posterior that is still poorly calibrated early in training. No free-bits threshold is applied, so the divergences are never clipped.

Reweighted emissions. The semantic heads in (18) use per-class weights. This is the standard remedy for the imbalance of BEV layouts, where vehicles and lane markings occupy a small fraction of the grid.

Discriminative terms. $\mathcal{L}_{\mathrm{plan}}$ and $\mathcal{L}_{\mathrm{gate}}$ supervise the planner head, which is not part of the generative model at all. The gate loss is a binary cross-entropy that carries per-expert positive weights offsetting how rarely each behavior occurs. $\mathcal{L}_{\mathrm{aux}}$ collects the remaining regularizers, which include the action-imitation term and the collision penalty on the planned waypoints. It carries no outer coefficient because each of its terms is already weighted individually. The overall objective is therefore a variationally motivated loss augmented with discriminative supervision, and $\lambda_{\mathrm{pred}}, \lambda_{\mathrm{plan}}, \lambda_{\mathrm{gate}}$ trade the two against each other. All weights appear in Appendix C.

A.8 Remarks on the Variational Family

Two restrictions on q_ϕ deserve comment. Both shrink the family without invalidating the bound, since (16) holds for any q_ϕ absolutely continuous with respect to the true posterior, and a smaller family only makes the slack larger.

First, the posteriors in (9) are filtering rather than smoothing distributions (Krishnan et al., 2015): each conditions on the past and the current observation but not on future observations, whereas the true posterior depends on the whole sequence. This is deliberate, because the same networks run at deployment without future frames, so training and closed-loop inference share one procedure.

Second, both encoders are frozen, which confines q_ϕ to distributions that factor through two fixed maps. In exchange, the cognitive feature is a fixed function of the images, making it cacheable and the asynchronous schedule implementable.

We have demonstrated that the training objective of DualWM follows from variational inference in a sequential latent-variable model with two temporal scales. The marginal likelihood of the recorded observations and maneuver annotations is bounded below by (14), whose cognitive sum accrues on the query set and whose behavioral sum accrues at frame rate, and whose divergences couple the two layers during learning. The implemented objective combines variational losses with discriminative supervision for the planner and a prior-rollout regularizer, and optimizes the result in two stages.

B Network Architectures

This section restates the model as a list of its conditional distributions and then specifies the networks that parameterize each of them. Layer widths are those of the released configuration, and every module described here is trained from scratch apart from the two frozen encoders.

B.1 Generative and Inference Distributions

The generative model consists of the following conditional distributions:

$$\mathbf{s}_0^h \sim p_\theta(\mathbf{s}_0^h) \qquad \mathbf{s}_0^l \sim p_\theta(\mathbf{s}_0^l \mid \mathbf{s}_0^h)$$
$$\mathbf{s}_t^h \sim p_\theta(\mathbf{s}_t^h \mid \mathbf{s}_{t-K}^h, \mathbf{u}_{t-K}, \mathbf{s}_{t-K:t-1}^l) \qquad \mathbf{s}_t^l \sim p_\theta(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h)$$
$$\mathbf{y}_t \sim p_\theta(\mathbf{y}_t \mid \mathbf{s}_t^h) \qquad \mathbf{o}_t \sim p_\theta(\mathbf{o}_t \mid \mathbf{s}_t^l).$$

The variational distribution, also called the inference model or the posterior, is represented by the following factorization:

$$\mathbf{s}_0^h \sim q_\phi(\mathbf{s}_0^h \mid \mathbf{c}_0) \qquad \mathbf{s}_0^l \sim q_\phi(\mathbf{s}_0^l \mid \mathbf{s}_0^h, \mathbf{o}_0)$$
$$\mathbf{s}_t^h \sim q_\phi(\mathbf{s}_t^h \mid \mathbf{s}_{t-K}^h, \mathbf{u}_{t-K}, \mathbf{s}_{t-K:t-1}^l, \mathbf{c}_t) \qquad \mathbf{s}_t^l \sim q_\phi(\mathbf{s}_t^l \mid \mathbf{s}_{t-1}^l, \mathbf{a}_{t-1}, \bar{\mathbf{s}}_t^h, \mathbf{o}_t).$$

The cognitive factors are instantiated on the query set $t \in \mathcal{T}_c$ and the behavioral factors at every frame $t = 0, \ldots, T$, with the initial distributions ($t = 0$) in place of the transitions. The initial distribution $p_\theta(\mathbf{s}_0^h)$ is a standard normal, and every other latent distribution is a diagonal Gaussian parameterized by a network. Each posterior carries exactly one evidence argument beyond its prior counterpart, $\mathbf{c}_t$ for the cognitive state and $\mathbf{o}_t$ for the behavioral one, and q_ϕ carries no emission factors because the emissions $\mathbf{y}_t, \mathbf{o}_t$ are observed. The product form used by the derivation, together with the head decomposition of the two emissions, appears as Eqs. (8)–(9) and (17)–(18).

B.2 Cognitive Layer

The cognitive layer is driven by the planning token $\mathbf{c}_t \in \mathbb{R}^{4096}$ that the frozen ORION (Fu et al., 2025) backbone produces from the surround-view images, together with the action chunk $\mathbf{u}_{t-K} \in \mathbb{R}^{K\times 2}$ executed since the previous query frame. Both are available only on $\mathcal{T}_c$.

The action chunk is encoded frame by frame into 64-dimensional embeddings and processed by the shared recurrent transition. The prior and the posterior use separate frame-level action encoders.

The planning token is compressed by a gated adapter rather than by a single projection. A token branch applies layer normalization and two fully connected layers of 1024 and 512 units with a GELU non-linearity in between. A context branch maps the concatenation of the recurrent state and the action embedding to 512 units. The two branches are concatenated and passed through a fusion network of 512 and 1024 units, whose output is split into a gate $\mathbf{g}$ and a residual $\boldsymbol{\delta}$. The adapter returns $\mathrm{LayerNorm}(\mathbf{z}_{\mathrm{tok}} + \mathrm{sigmoid}(\mathbf{g}) \odot \boldsymbol{\delta})$. Compressing 4096 dimensions to 512 by a single linear map discards information without regard to the current situation. The gate instead lets the recurrent state select which components of the token are admitted, which keeps the adapter conditioned on what the vehicle already believes.

The recurrent path applies a linear layer of 512 units with a leaky ReLU non-linearity to the previous stochastic state and updates a gated recurrent unit of 512 units. The mean of the K behavioral states over the preceding query interval is concatenated with the last state in that interval and projected to a 64-dimensional feedback feature. The prior reads the concatenation of the recurrent state, action embedding, and feedback feature, which is 640 dimensions. The posterior also reads the 512-dimensional adapter output, for 1152 dimensions.

Both distributions are produced by the same network form. A fully connected layer whose width equals its input is followed by a leaky ReLU non-linearity and a linear layer emitting the mean together with a pre-transformed standard deviation $\tilde{\sigma}$. The floor of Eq. (20) is realized as

$$\sigma = 2\,\mathrm{sigmoid}(\tilde{\sigma}/2) + \sigma_{\min}, \qquad \sigma_{\min} = 0.1, \tag{26}$$

which confines σ to $(0.1, 2.1)$ and bounds the logarithm in Eq. (20) from both sides. The behavioral layer uses the identical form.

During cognitive pretraining, the stochastic state passed to the next transition is selected from the prior with probability 0.15 after the first query step, following Hu et al. (2022a). During joint training, the cognitive transition uses prior means for the first $K - 1$ steps of each action chunk and a posterior sample at the query frame. The behavioral layer uses a fixed cognitive state between queries.

The heads that emit $\mathbf{y}_t$ read the cognitive state through a shared input. The route geometry ahead is first encoded by layer normalization and two fully connected layers of 64 units. Its input is a sequence of six points sampled at 2 to 12 meters along the reference path, each with its planar coordinates and a validity flag, giving 18 dimensions. Concatenating this embedding with the 1024-dimensional cognitive state gives a 1088-dimensional head input. The lateral and longitudinal heads apply a causal one-dimensional convolution with kernel size five over the state sequence, followed by layer normalization, two fully connected layers of 256 units with GELU non-linearities, and dropout 0.1. The convolution is causal, so a prediction at one step reads its own past but no future

step. Distinguishing a turn from a lane change requires observing how the state evolves, which a per-step network cannot represent. The six recognition heads share a single network that reads one step at a time, consisting of layer normalization followed by fully connected layers of 512 and 256 units with GELU non-linearities and dropout 0.1.

B.3 Behavioral Layer

The observation $\mathbf{o}_t^{\mathrm{bev}}$ enters as a feature map with 64 channels on a 160×160 grid, produced by the frozen BEVFormer (Li et al., 2022) encoder from six surround-view cameras and cached offline. A stride-two convolution raises the channel count to 128, and three residual blocks with stride two carry it to 256 while reducing the grid to 10×10. Adaptive average pooling brings the grid to 4×4 rather than to a single vector, so coarse spatial layout survives the projection. The result is flattened and projected to 512 dimensions. The scalar speed is normalized by $12\,\mathrm{m/s}$ and encoded by two fully connected layers of 128 units. Their concatenation is projected back to a 512-dimensional observation embedding.

The recurrent path mirrors the cognitive one at a larger width. A linear layer of 1024 units with a leaky ReLU non-linearity precedes a gated recurrent unit of 1024 units. The frame action $\mathbf{a}_{t-1} \in \mathbb{R}^2$ is encoded to 64 dimensions by a linear layer with a leaky ReLU non-linearity, again with separate encoders for the prior and the posterior. The prior reads the recurrent state, the action embedding, and the held cognitive state, which is 2112 dimensions. The posterior additionally reads the observation embedding, which is 2624 dimensions. Posterior replacement with probability 0.15 is applied as in the cognitive layer.

Every head in (18) is decoded by a transposed convolutional network of one shared design. A linear layer maps the 1536-dimensional behavioral state to 512 channels on a 1×1 grid. A transposed convolution expands this to a constant grid, and three stride-two transposed convolutions of 512 channels follow. Three further stride-two stages reduce the channel count to 256, 128, and 64, each carrying its own output head, so every emission is supervised at three resolutions. The constant grid is 6×6 for the BEV heads, giving outputs at 96×96, 192×192, and 384×384, and 3×6 for the front-view heads, giving 48×96, 96×192, and 192×384. The BEV targets are rendered at $0.2\,\mathrm{m}$ resolution over a $\pm 40\,\mathrm{m}$ range and decoded at the 384×384 output grid, so the behavioral state is held responsible for the full surround, not only the forward-facing part.

Among the BEV heads, the path channel deserves a note because it is not a perception target in the usual sense. Its first component is a corridor of the vehicle's width swept by the expert future trajectory, and its second holds the time at which the vehicle reaches each cell of that corridor. Since the path channel is decoded from the same state as the semantic layout, one representation is supervised by both what the scene contains and where the expert intends to go.

B.4 Cognition-Behavior Coupled Planner

The planner reads the joint state $\mathbf{s}_t = [\bar{\mathbf{s}}_t^h; \mathbf{s}_t^l] \in \mathbb{R}^{2560}$ together with a twelve-dimensional target point formed from the same six route points without the validity flags. A base decoder receives this input augmented with the lateral and longitudinal intent probabilities, giving a 2581-dimensional input, and predicts six future positions spaced half a second apart, covering three seconds ahead. Positions are expressed in the vehicle-centered LiDAR frame, whose two axes point to the right and forward. The decoder consists of fully connected layers of 1024, 512, 256, and 64 units with GELU non-linearities and dropout 0.2.

Four experts share the base input and additionally read the six recognition probabilities, giving a 2587-dimensional input. Each expert is a narrower network of the same depth with a hidden width of 128, and each predicts a residual over the same six positions. The four experts correspond to four driving behaviors: braking, bypassing, merging, and turning. Braking is a deceleration to a stop, bypassing a lateral pulse that returns to the original lane, merging a lateral step that stays in the new lane, and turning at a junction turn. These four cover the main non-trivial maneuvers that recur in the data, while ordinary straight cruising is handled by the base trajectory and needs no expert, so no further experts are added beyond these classes. The router applies layer normalization and two fully connected layers of 64 units to the fifteen-dimensional concatenation of the lateral, longitudinal, and recognition probabilities, and emits four logits whose sigmoids give the expert

Algorithm 1 Two-stage training of DualWM

Require: dataset $\mathcal{D}$; frozen VLM f_{vlm}; frozen BEV encoder f_{bev}; query interval K

Stage 1: cognitive layer
for each minibatch of clips in $\mathcal{D}$ **do**
  On $t \in \mathcal{T}_c$, extract $\mathbf{c}_t = f_{\mathrm{vlm}}(\cdot)$ and update $\mathbf{s}_t^h$
  Decode intents from $\mathbf{s}_t^h$
  Update the cognitive parameters by $\mathcal{L}_{\mathrm{int}} + \mathcal{D}_t^h$
end for
Freeze the intent heads and cache cognitive states for window initialization
Stage 2: joint optimization
for each minibatch of clips in $\mathcal{D}$ **do**
  Extract $\mathbf{o}_t = f_{\mathrm{bev}}(\cdot)$ and initialize $\mathbf{s}_0^h$ from the cache
  Update $\mathbf{s}_t^h, \mathbf{s}_t^l$ via Eq. (9), reconstruct the current BEV, and roll out future states
  Predict the base trajectory and intent-gated expert residuals
  Update the remaining parameters by $\mathcal{L}$
end for

weights $g_{m,t}$. Here, behavior-grounded intent refinement refers to the feedback path $\mathbf{s}_{t-K:t-1}^l \rightarrow \mathbf{s}_t^h \rightarrow$ intent probabilities at query frames. The behavioral-state history conditions the cognitive update, and the intent heads decode probabilities from the updated cognitive state. At every control step, the current behavioral state also conditions the base trajectory and expert residuals through the joint state.

An action decoder predicts the immediate control pair alongside the trajectory, supplying the action-imitation term collected in $\mathcal{L}_{\mathrm{aux}}$. It reads the joint state without the target point, which is 2560 dimensions, and consists of three fully connected layers of 2560, 2560, and 1280 units with ReLU non-linearities followed by a linear layer with a tanh non-linearity, which confines steering and acceleration to $[-1, 1]$. The executed control does not come from this decoder. At deployment the planned trajectory is converted to steering and throttle by a PID controller, and the state update is conditioned on the control that was actually executed. The action decoder instead keeps the behavioral state predictive of the immediate control rather than of the trajectory alone.

Three properties of this construction deserve comment. The router reads the intent probabilities and not the joint state. Routing is a decision about which behavior the situation calls for, and the cognitive layer is where that decision is represented. Admitting the 2560-dimensional state would let the router rediscover its own routing signal from behavioral features and dilute the supervision the intent heads provide. The four weights are independent sigmoids rather than a softmax over experts. Several behaviors co-occur, as when a turn is executed under a stop sign, and an ordinary frame requires every weight to be near zero, neither of which a softmax can express. The planner forms its output as $\boldsymbol{\tau}_t = \boldsymbol{\tau}_{\mathrm{base},t} + \sum_m g_{m,t} \boldsymbol{\Delta}_{m,t}$. The base trajectory enters this sum with its gradient detached, so the combined objective updates the gates and the residuals while leaving the base to its own trajectory loss. The residuals therefore learn to correct a fixed base rather than to reshape it. On an ordinary frame the gated residuals stay near zero, so the base governs the output.

C Training and Evaluation Details

C.1 Algorithms

Algorithm 1 summarizes the two-stage training procedure and Algorithm 2 the asynchronous closed-loop inference. We adopt two stages because the cognitive layer is driven by a frozen vision-language model whose features are expensive to recompute. The cognitive layer and its intent heads are trained first. Both layers are then jointly optimized with the behavioral decoders and MoE planner under the remaining loss terms, while the intent heads remain frozen. At deployment, the vision-language model is queried only on the low-frequency set $\mathcal{T}_c = \{0, K, 2K, \dots\}$. Between queries the most recent cognitive state is held, while the behavioral layer and the planner update at every control step.

Algorithm 2 Asynchronous closed-loop inference

Require: query interval K; controller π

$\mathbf{c}_0 \leftarrow f_{\mathrm{vlm}}(\cdot)$ and $\mathbf{o}_0 \leftarrow f_{\mathrm{bev}}(\cdot)$; initialize $\mathbf{s}_0^h$ from $\mathbf{c}_0$ and $\mathbf{s}_0^l$ from $\mathbf{s}_0^h, \mathbf{o}_0$; set $\bar{\mathbf{s}}_0^h \leftarrow \mathbf{s}_0^h$
$\boldsymbol{\tau}_0 \leftarrow \boldsymbol{\tau}_{\mathrm{base},0} + \sum_m g_{m,0} \boldsymbol{\Delta}_{m,0}$ and execute $\mathbf{a}_0 \leftarrow \pi(\boldsymbol{\tau}_0)$
for each control step $t = 1, 2, \ldots$ **do**
  if $t \in \mathcal{T}_c$ **then**
    $\mathbf{c}_t \leftarrow f_{\mathrm{vlm}}(\cdot)$ and update $\mathbf{s}_t^h$ from $\mathbf{s}_{t-K}^h$, $\mathbf{u}_{t-K}$, $\mathbf{s}_{t-K:t-1}^l$, and $\mathbf{c}_t$
    $\bar{\mathbf{s}}_t^h \leftarrow \mathbf{s}_t^h$ {refresh the held state}
  else
    $\bar{\mathbf{s}}_t^h \leftarrow \bar{\mathbf{s}}_{t-1}^h$ {hold the cognitive state}
  end if
  $\mathbf{o}_t \leftarrow f_{\mathrm{bev}}(\cdot)$ and update $\mathbf{s}_t^l$ from $\mathbf{s}_{t-1}^l$, $\mathbf{a}_{t-1}$, $\bar{\mathbf{s}}_t^h$, and $\mathbf{o}_t$
  $\boldsymbol{\tau}_t \leftarrow \boldsymbol{\tau}_{\mathrm{base},t} + \sum_m g_{m,t} \boldsymbol{\Delta}_{m,t}$
  $\mathbf{a}_t \leftarrow \pi(\boldsymbol{\tau}_t)$ and execute $\mathbf{a}_t$
end for

C.2 Dataset and Baselines

We evaluate on Bench2Drive (Jia et al., 2024), a closed-loop benchmark built on the CARLA simulator (Dosovitskiy et al., 2017). Following all baselines, we adopt the official *base* training set of 1000 clips, each covering roughly 150 m of continuous driving in a single traffic scene, and split it into 950 clips for training and 50 for validation. Closed-loop evaluation uses the official 220 short routes that cover 44 interactive scenarios with five routes each. We adopt the base setting rather than a larger split so that all methods are compared under an identical data budget, which isolates the effect of the model design from that of training-data scale.

We compare DualWM against methods that span the main paradigms of closed-loop driving. Among classical imitation-learning planners, TCP (Wu et al., 2022) couples a trajectory branch with a control branch and lets the trajectory prediction guide low-level control, and its trajectory-only variant is denoted TCP-traj. VAD (Jiang et al., 2023) represents the scene with a vectorized map and agent elements and plans efficiently over this compact representation. UniAD (Hu et al., 2023b) integrates detection, tracking, mapping, motion forecasting, occupancy prediction, and planning into a single query-based planning-oriented architecture. ThinkTwice (Jia et al., 2023b) adopts a scalable coarse-to-fine decoder and distills features from a privileged expert. DriveTransformer (Jia et al., 2025) unifies the pipeline with sparse queries that attend directly to multi-view sensor features. DriveAdapter (Jia et al., 2023a) connects a trainable perception network to a frozen privileged driving policy through adapters.

Two baselines learn a policy beyond pure imitation. Raw2Drive (Yang et al., 2025) is a model-based reinforcement-learning framework that first trains a privileged world model together with a neural planner and then aligns a raw-sensor world model to it through a guidance mechanism. ReCogDrive (Li et al., 2026) couples a cognitive vision-language model with a generative diffusion planner and refines the policy by reinforcement learning, which avoids the infeasible actions and slow inference of methods that emit trajectories in the language space.

The remaining baselines introduce a vision-language backbone. ORION (Fu et al., 2025) maps the reasoning of a vision-language model to planning tokens and generates trajectories from them, and we adopt it as the frozen cognitive backbone of DualWM. DriveMoE (Yang et al., 2026) builds a vision-language-action model with a scene-specialized vision mixture-of-experts that dynamically selects relevant cameras and a skill-specialized action mixture-of-experts that activates behavior-specific modules to avoid mode averaging. MindDrive (Fu et al., 2026) is a vision-language-action model trained by online reinforcement learning, using a single language model with separate decision and action experts that map linguistic decisions into feasible trajectories. LVDrive (Mei et al., 2026) augments a vision-language-action model with a future-scene prediction task learned in a latent space under supervision from a frozen vision backbone, modeling future scene and motion jointly in a single forward pass rather than by autoregressive generation.

Baseline driving-performance results are taken from the corresponding publications and the official Bench2Drive leaderboard.

Data preprocessing. From the raw Bench2Drive clips, we precompute the inputs and supervision targets of the two layers. For the behavioral layer, a frozen BEVFormer encodes the surround-view images into a bird's-eye-view embedding that we compress and cache offline, and the layer is supervised with bird's-eye-view semantic, instance, and flow maps rendered from the simulator annotations. For the cognitive layer, we derive lateral, longitudinal, and safety-related intent labels from the expert trajectory and surrounding scene state, with thresholds calibrated on expert behavior.

C.3 Evaluation Metrics

We report the five official Bench2Drive metrics over all evaluation routes.

Driving Score (DS) averages the product of route completion and the infraction factor,

$$\mathrm{DS} = \frac{1}{R}\sum_{r=1}^{R} C_r\, P_r, \tag{27}$$

where $C_r \in [0, 100]$ is the completed percentage of route r and $P_r \in [0, 1]$ is its cumulative infraction penalty.

Success Rate (SR) is the fraction of routes completed without any infraction other than a minimum-speed violation,

$$\mathrm{SR} = \frac{1}{R}\sum_{r=1}^{R} \mathbf{1}[\text{route } r \text{ completed } \wedge \text{ no major infraction}]\,. \tag{28}$$

Efficiency is the mean minimum-speed percentage over the routes that record a minimum-speed infraction, which measures how well the ego vehicle maintains progress relative to surrounding traffic.

Comfortness averages over all routes the fraction of trajectory segments whose per-step kinematics stay within predefined comfort bounds. The bounded quantities are the longitudinal and lateral acceleration, the jerk, the yaw rate, and the yaw acceleration.

Multi-Ability reports the success rate within each of five scenario groups: Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign. The first four use the same success criterion as SR. Traffic Sign additionally requires the ego vehicle to pass the relevant junction without a stop-sign or red-light violation, following the official ability benchmark.

C.4 Hyperparameters

Table 5 reports the full configuration. We employ a frozen ORION (Fu et al., 2025) as the cognitive backbone, queried every $K = 10$ control steps, and a frozen BEVFormer (Li et al., 2022) as the observation encoder. The planner predicts six waypoints over a three-second horizon and combines a base trajectory with four intent-gated behavior experts. We optimize the objective with AdamW (Loshchilov & Hutter, 2019) at a learning rate of 1×10^{-4} under a cosine-annealing schedule. Per-head BEV reconstruction and per-expert gate weights follow our released configuration.

We did not perform an extensive hyperparameter sweep. Most values follow common practice and our released configuration. The query interval $K = 10$ corresponds to one second of simulated driving at the 10 Hz simulation frame rate. For DualWM, we report end-to-end latency as the mean wall-clock time per control step over the complete online pipeline at $K = 10$, including image decoding, BEV encoding, online VLM inference, world-model updates, and trajectory planning. The VLM runs once every ten control steps, and its inference cost contributes to the average over all control steps. Under this protocol, DualWM achieves an average latency of 551 ms per step, compared with 1493 ms for ORION on the same hardware, a reduction of 63.1%. The two layers operate at different update frequencies: the cognitive layer updates every K steps, while the behavioral layer updates at each step. The behavioral layer performs latent-state computation on BEV representations; the end-to-end latency also includes the cost of producing those representations and invoking the VLM. The 10 Hz simulation rate specifies the temporal spacing of driving observations and state transitions, independently of wall-clock inference time. Together with the closed-loop results, the latency comparison establishes a better trade-off between driving performance and inference cost, representing

Table 5: Full hyperparameter configuration of DualWM.

Hyperparameter	Value
Model	
Query interval K	10
Training sequence length	10
Behavior experts (4)	Brake/Bypass/Merge/Turn
Waypoints $\times$ horizon	6, 3 s
Cognitive state $\mathbf{s}^h$ (deter/stoch)	512 / 512
Behavioral state $\mathbf{s}^l$ (deter/stoch)	1024 / 512
VLM feature $\rightarrow$ projection	4096 $\rightarrow$ 512
BEV embedding dim	512
Cognitive-context dim	1024
Action latent dim	64
Dropout (intent/planner)	0.1 / 0.2
Optimization	
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	0.01
LR schedule	Cosine ($\eta_{\min} = 10^{-6}$)
Gradient clipping (max norm)	5.0
Effective batch size	16
Training epochs	50 (early stopping, patience 10)
Loss weights	
Balanced-KL α	0.75
KL λ_{KL}	1×10^{-3}
Trajectory λ_{plan} (gated / base)	6.0 / 6.0
Gate λ_{gate}	2.0
Prediction λ_{pred} (+1/+2/+3 s)	2.0 / 2.0 / 1.0
Action imitation	4.0
Collision penalty	5.0
BEV reconstruction / path	6.0 / 1.0
Front semantics / depth	3.0 / 10.0
Random seeds	$0, 1, 2$

progress toward real-time deployment. The loss weights emphasize multi-step BEV prediction and gating over the near-trivial one-step prior. The planner adopts a residual mixture-of-experts head rather than a single shared head, since a single shared head underperformed on demanding maneuvers. All hyperparameters were selected on the validation split, with no evaluation-route information used.

C.5 Computational Infrastructure

All experiments were run on a single NVIDIA A800-80GB GPU paired with an Intel Xeon Platinum 8374C CPU and 128 GB of RAM. The software stack is Python 3.8 and PyTorch 2.4.1 with CUDA 11.8 and cuDNN 9.1, and closed-loop evaluation uses CARLA 0.9.15. One training epoch takes about three hours. Each closed-loop evaluation over 220 routes takes about two days per seed.

D Additional Experiments

D.1 Ablation Studies

We report four ablations of the full model, each removing one component or coupling pathway. The *w/o cognition* variant removes the cognitive conditioning, so the behavioral state evolves on BEV observations and past actions alone and the planner receives no intent gating. The *w/o feedback*

variant removes the bottom-up behavioral-state feedback $\mathbf{s}^l_{t-K:t-1}$ from the cognitive update, while retaining the top-down cognitive conditioning of the behavioral layer. The *w/o prediction* variant disables predictive modeling in both layers while retaining the backbone encoders, cognitive conditioning, temporal encoding, and MoE planner. The *w/o MoE* variant replaces the intent-gated mixture-of-experts planner with a single shared trajectory head over the same joint state. Every variant is evaluated over the same 220 routes.

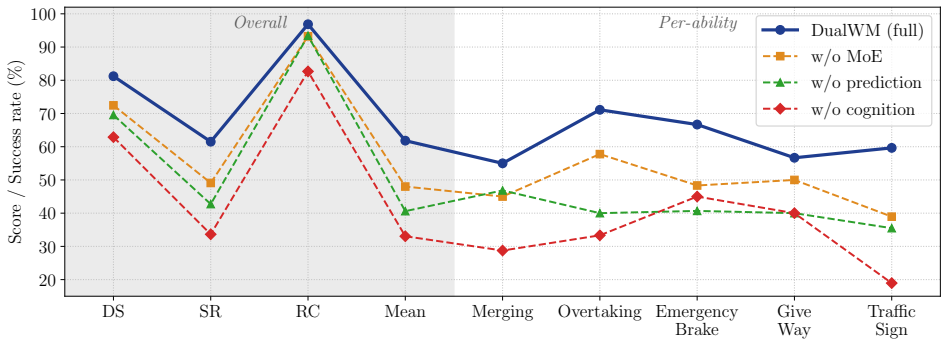


Figure 5: Ablation of DualWM on the overall metrics (driving score, success rate, route completion, mean ability) and the five per-ability success rates. The per-ability differences are read as trends rather than exact gaps, and the full model leads throughout.

Figure 5 separates each ablation across the overall metrics and the five abilities that the aggregate driving score cannot distinguish. Route completion stays high for every variant, so the degradation appears in the driving score, the success rate, and the abilities rather than in how far the vehicle drives. The way the loss divides among the abilities identifies what each component contributes. Table 6 reports the corresponding per-ability success rates.

Removing cognition degrades every ability at once. The loss ranges from 16.67 points on Give Way to 40.70 points on Traffic Sign, and no ability is spared. This across-the-board collapse is the reason the cognitive ablation records the largest aggregate loss, since a behavioral layer without a maneuver-level goal reacts to the immediate scene in every situation rather than only in a few.

Removing prediction lowers every ability that depends on forecasting how other agents will move. Overtaking drops by 31.11 points, Emergency Brake by 25.99, and Traffic Sign by 24.17, while Merging, which is settled more by the intent to merge than by a precise forecast, moves by only 8.16 points. Anticipating the motion of the lead and oncoming vehicles is what the prior rollout supplies, so the abilities that need this anticipation are the ones it most affects. The corresponding cognitive ablation lowers Merging by 26.25 points, which is consistent with merging being governed by intent rather than by forecasting.

Among the three variants with per-ability results, removing the MoE planner yields the smallest DS decrease, because the base trajectory survives and only the behavior-specific residuals are removed. Its loss spreads across the maneuver abilities without concentrating on one, from 6.67 points on Give Way to 20.70 points on Traffic Sign, as a single shared head reabsorbs the behaviors that the four experts had separated. Across the three variants the losses therefore occupy different regions of the ability space rather than a shared one, so the components act in complementary roles that the pooled metrics conceal.

D.2 Query-Interval Sensitivity

With model parameters held fixed, we evaluate sensitivity to the query interval over $K \in \{6, 8, 10, 12, 14\}$ using an earlier checkpoint of the same model. Only K varies; all other evaluation settings remain unchanged. We completed this closed-loop sweep before selecting the final checkpoint used for the main results. Thus, the DS of 76.75 at $K = 10$ and the main-result DS of 81.22 correspond to different checkpoints. As shown in Table 7 and Figure 6, $K = 10$ achieves the highest observed DS among the tested settings for the earlier checkpoint.

Table 6: Per-ability success rates (%) of the ablation variants on Bench2Drive.

Ability	w/o cognition	w/o prediction	w/o MoE
Overtaking	33.33	40.00	57.78
Merging	28.75	46.84	45.00
Emergency Brake	45.00	40.68	48.33
Give Way	40.00	40.00	50.00
Traffic Sign	18.95	35.48	38.95
Mean	33.21	40.60	48.01

Table 7: Effect of the query interval K on driving score and inference latency.

K	DS↑	Latency (ms)↓
6	74.16	623
8	74.88	576
10	**76.75**	551
12	72.51	534
14	71.23	**520**

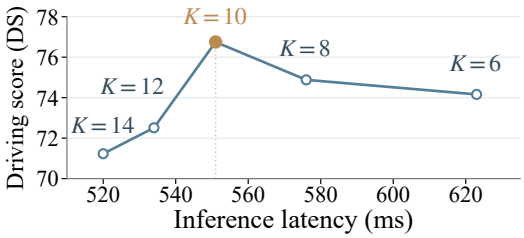


Figure 6: Driving score versus inference latency across different query intervals K.

Table 8: Per-seed results of DualWM on Bench2Drive. Std is the sample standard deviation over the three seeds.

Metric	Seed 0	Seed 1	Seed 2	Mean ± Std
Driving Score	81.60	81.33	80.74	81.22 ± 0.44
Success Rate (%)	62.73	61.36	60.45	61.51 ± 1.15
Route Completion	97.17	96.77	96.63	96.86 ± 0.28
Infraction Penalty	0.831	0.832	0.825	0.829 ± 0.004
Efficiency	179.01	174.91	177.38	177.10 ± 2.06
Comfortness	22.53	20.37	23.24	22.05 ± 1.49
Merging (%)	57.50	53.75	53.75	55.00 ± 2.17
Overtaking (%)	68.89	71.11	73.33	71.11 ± 2.22
Emergency Brake (%)	70.00	68.33	61.67	66.67 ± 4.41
Give Way (%)	60.00	50.00	60.00	56.67 ± 5.77
Traffic Sign (%)	60.00	61.05	57.89	59.65 ± 1.61
Ability Mean (%)	63.28	60.85	61.33	61.82 ± 1.29

D.3 Per-Seed Results and Variance

Table 8 reports every metric for each random seed with the mean and the sample standard deviation across seeds ($n = 3$, $\mathrm{ddof} = 1$). Each seed is one closed-loop pass over the 220 routes with the traffic-manager seed set to 0, 1, and 2. The across-seed variation is small for the aggregate metrics and larger for the abilities defined on few routes, whose per-seed scores are more variable.

E Reconstructions and Predictions

We visualize the reconstructions and predictions of the behavioral layer, all decoded from the trained DualWM model of Appendix C on the validation split. A bird's-eye-view prediction is produced by rolling the behavioral prior forward without access to any observation after the current frame and decoding the semantic layout at each step. In every BEV panel, the colors denote road (gray), lane marking (white), vehicle (blue), pedestrian (red), static obstacle (green), and cyclist (yellow), over a black background, on a $\pm 40\,\mathrm{m}$ grid centered on the ego vehicle, with its heading up.

E.1 Decoded BEV Predictions

Comparing the reconstruction at t with the prediction at $t{+}1$ s isolates what the model forecasts rather than copies. In the first four scenes, the predictions differ from the current reconstructions, with changes in road geometry and traffic layout over the one-second horizon. In the last two scenes the scene barely changes over one second, and the prediction stays close to the reconstruction, which is the correct behavior when nothing moves. Pixel accuracy is between 0.88 and 0.97 for

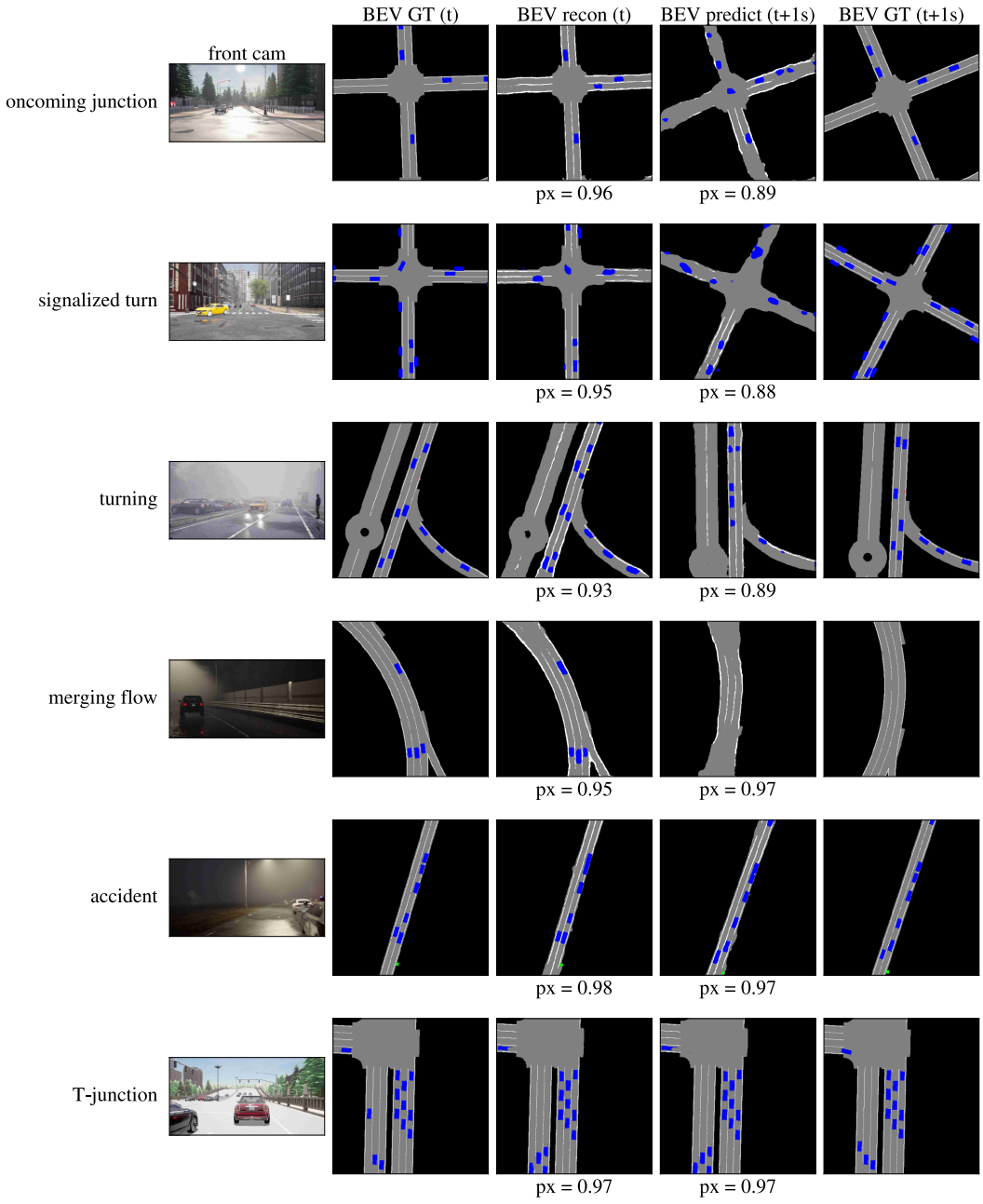


Figure 7: Decoded BEV predictions across six scenes. The first four rows undergo substantial change within the one-second horizon. The last two rows are near-stationary. Each row shows, left to right, the front camera at frame t, the BEV ground truth at t, the posterior reconstruction at t, the one-second prior prediction, and the BEV ground truth at $t{+}1$ s. The prior receives no observation after frame t. Pixel accuracy is annotated below each reconstruction and prediction.

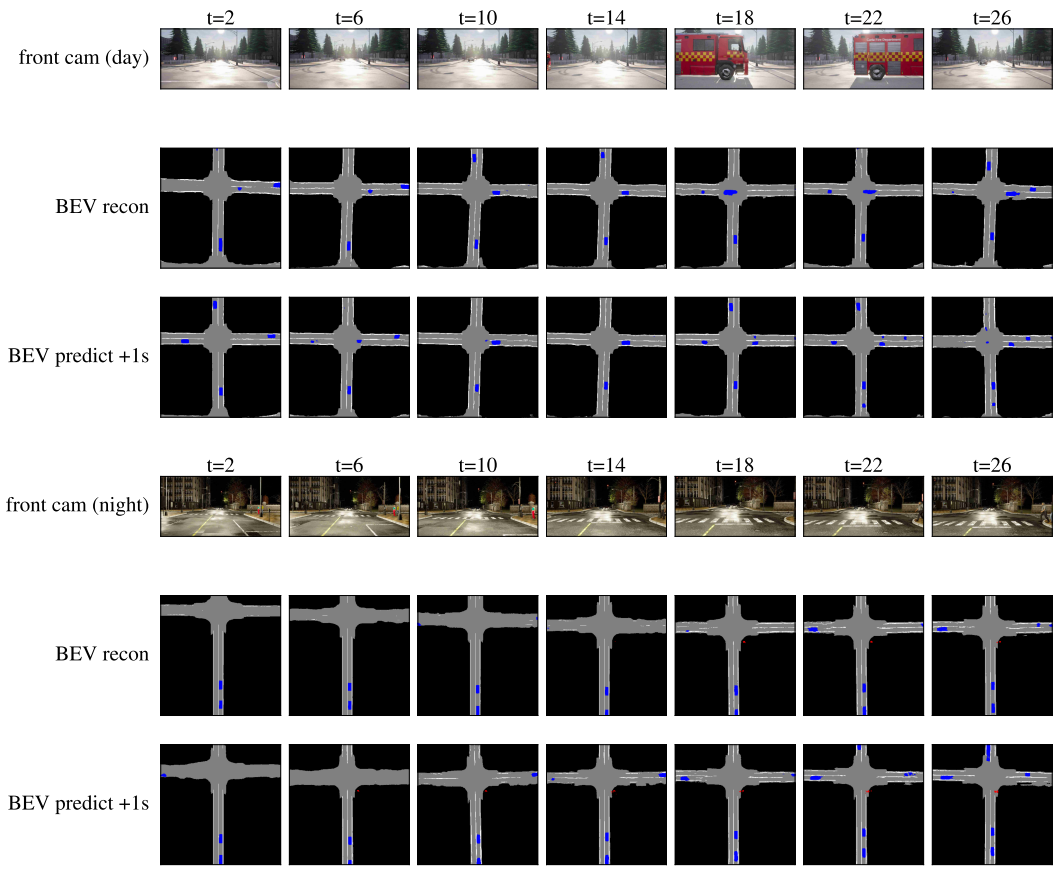


Figure 8: Camera and BEV over time along two routes through four-way junctions, a daytime crossing (top three rows) and a night crossing (bottom three rows). Columns are frames at $0.4\,\mathrm{s}$ spacing. Within each route, the three rows are the front camera, the posterior reconstruction, and the one-second prior prediction.

the prediction and between 0.93 and 0.98 for the reconstruction. The prediction accuracy sits a few points below the reconstruction accuracy in most scenes, consistent with the additional difficulty of forecasting a second ahead rather than encoding the observed frame, while the predicted layouts remain legible. A one-step prior spans $0.1\,\mathrm{s}$ and is near-trivial, so the one-second horizon obliges the latent state to represent how the scene evolves, not just its current state.

E.2 Camera and BEV Sequences

On the first route, a fire truck enters from the left arm of the junction (frames $t{=}18$ and $t{=}22$) and its bird's-eye footprint appears in both the reconstruction and the one-second prediction. On the second route, cross-traffic and a pedestrian enter the junction over the sequence and are likewise captured. In both routes the one-second prediction at each frame tracks the entering agents rather than reproducing the earlier empty junction, and the reconstruction and prediction rows stay close across the sequence. Because each column is the one-second prediction anchored at that frame, the match holds not at an isolated step but as the ego advances through the junction. The second route is captured after dark, and its predictions track the entering agents as closely as in the daytime route.

E.3 Front-view Reconstruction

The bird's-eye-view projection collapses the vertical dimension of the scene onto the ground plane, so object height and elevation are absent from the BEV layout alone. Reconstructing the front-view semantics and depth from the same behavioral state restores this missing dimension: pixel accuracy

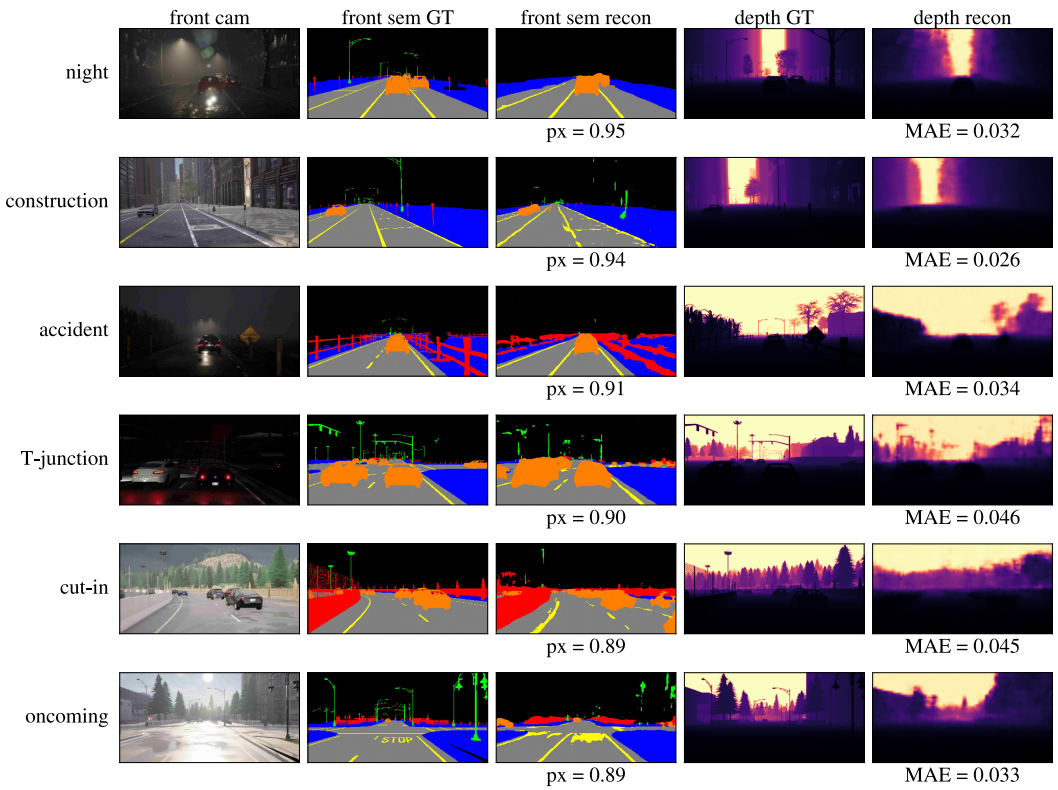


Figure 9: Front-view reconstruction across six scenes (a foggy night street, a daytime construction site, a foggy accident scene, a night junction with queued traffic, a daytime highway cut-in, and a foggy oncoming-traffic street). Each row shows the front camera, the eight-class front semantic ground truth, the posterior semantic reconstruction, the depth ground truth, and the depth reconstruction. The pixel accuracy and the mean absolute depth error are annotated below the reconstructions.

is between 0.89 and 0.95, and the depth error is between 0.026 and 0.046 across the six scenes, and the front-view depth in particular recovers the per-pixel range and the vertical structure that the top-down view discards. The behavioral latent is therefore supervised to encode the scene at two complementary levels, which are the metric ground-plane layout that the planner consumes and the front-view appearance that retains height, rather than either one alone. The two night scenes lie at opposite ends of this range. The foggy night street reaches a semantic accuracy of 0.95 and a depth error of 0.032. The queued-traffic night junction reaches a semantic accuracy of 0.90 and a depth error of 0.046. Within these two night scenes, low illumination alone does not appear to set the reconstruction quality. The queued vehicles and the multiple oncoming headlights in the second scene more plausibly explain its larger depth error. The first night scene contains a single light source and an otherwise empty road and shows no comparable degradation.

Across the three figures, the same behavioral state supports both a faithful reconstruction of the current frame and a one-second prediction that departs from it toward the future ground truth. The prediction and reconstruction accuracies differ by only a few points. These qualitative examples suggest that the state the planner consumes carries a usable forecast of how the scene will evolve, the property a world model must supply for anticipatory planning and what the prediction term of Appendix A is designed to induce.