MORE: LEARNING MOTION-ORIENTED REGISTERS FOR DRIVING WORLD MODELS AND PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

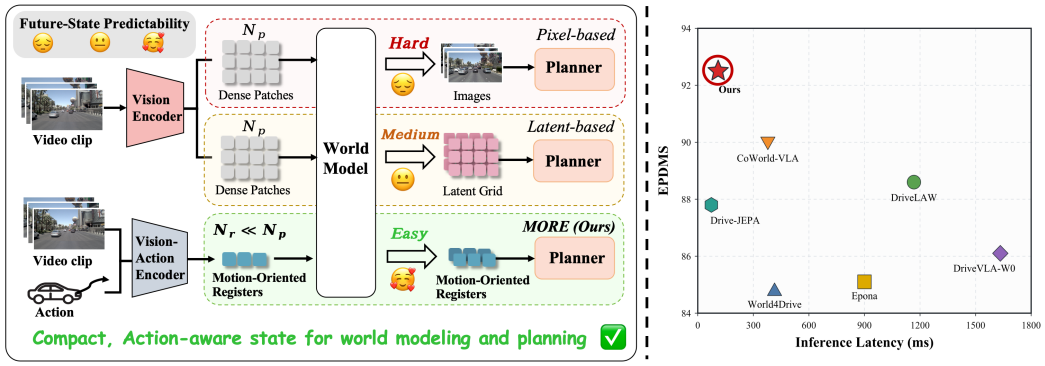
By predicting future states, latent world models provide rich self-supervision for end-to-end driving, but often inherit dense features from pretrained visual encoders as their world-state interface. This creates a mismatch: visual pretraining produces information-rich, spatially dense features, but their adaptation for motion-conditioned world modeling and planning is deferred to downstream training. The dense grid also makes future prediction more computationally expensive as visual resolution increases. We therefore argue that video pretraining should directly learn state representations suited to downstream world modeling and planning, rather than relying on downstream training to adapt inherited visual features. We propose ***MORE***, a V-JEPA framework that learns **M**otion-**O**riented **RE**gisters during video pretraining as a shared state interface for driving world models and planning. MORE injects temporally aligned ego motion into a small set of registers and uses them as the sole context for masked patch prediction, channeling dense patch-level supervision through a compact state bottleneck. A complementary blockwise register-masking objective further shapes predictive structure in the register space. During downstream training, the same register space defines the future states predicted by the world model and provides shared scene memory for trajectory generation and scoring. On NAVSIM-v2, MORE outperforms the state of the art by 1.6 in EPDMS, while achieving superior efficiency with only 1/10 the inference latency of DriveLaW. These results support learning the state interface for world modeling and planning directly during video pretraining.

1 INTRODUCTION

End-to-end autonomous driving (Wu et al., 2022; Hu et al., 2023; Jiang et al., 2023; Sun et al., 2025) predicts future trajectories from sensor observations, but low-dimensional expert trajectories provide limited supervision for high-capacity visual encoders. Driving world models provide richer signals by predicting future observations (Wang et al., 2024b; Li et al., 2026c; Wang et al., 2024a), occupancy or BEV states (Zheng et al., 2024; Yang et al., 2025; Zhang et al., 2026), or latent features (Li et al., 2025b; Zheng et al., 2025). Joint-Embedding Predictive Architectures (JEPA) (LeCun, 2022) offer a complementary route by learning predictable representations rather than reconstructing pixels. V-JEPA (Bardes et al., 2024) extends this principle to spatiotemporal masked prediction in videos, while V-JEPA 2-AC (Assran et al., 2025) and Drive-JEPA (Wang et al., 2026b) further demonstrate its potential for action-conditioned prediction and trajectory planning.

However, a fundamental mismatch remains between pretrained visual representations and the states required by downstream world models and planners. Visual pretraining primarily learns information-rich representations of scene content and spatiotemporal structure. In contrast, a world model requires states that capture scene evolution conditioned on ego motion and support future prediction (Min et al., 2024), while a planner further requires decision-relevant context. Existing approaches typically learn visual representations first and adapt them during downstream training (Yang et al., 2026a; Zheng et al., 2025; Li et al., 2025b). For example, V-JEPA 2-AC (Assran et al., 2025) learns action-conditioned dynamics over frozen visual features after action-free pretraining. LAW (Li et al., 2025b) introduces ego-trajectory-conditioned future latent prediction during end-to-end training, while WorldRFT (Yang et al., 2026a) further aligns scene representations with planning through geometric priors and planning supervision. Together, these methods indicate that rich visual features do not automatically form ego-motion-sensitive world states suited to future

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.



(a) State-Interface Mismatch and Our Design. **(b) Latency and Performance Comparison.**

Figure 1: **State-interface mismatch and inference efficiency.** (a) Comparison of dense state interfaces and MORE's compact motion-oriented register interface for world modeling and planning. (b) EPDMS versus inference latency.

prediction, leaving much of the structure for state evolution and control to be learned downstream. We refer to this gap, illustrated in Fig. 1(a), as a ***state-interface mismatch***.

Beyond this alignment gap, the organization of future states poses a related efficiency challenge. V-JEPA 2-AC (Assran et al., 2025) predicts spatiotemporally indexed patch features, DriveX (Shi et al., 2025) and WoTE (Li et al., 2025c) predict latent BEV grids, and World4Drive (Zheng et al., 2025) predicts spatial visual latents. Because these future states retain spatial grids, finer grids require more tokens to be predicted at each future step. DrivoR (Kirby et al., 2026) and NTR (Li et al., 2026a) instead compress dense visual features into a few planning tokens. This reduces state size for current-scene planning, but compactness alone does not ensure ego-motion sensitivity or the predictive structure needed for future modeling.

Our central question is therefore not how to further compress visual features during downstream training, but whether video pretraining can directly learn state representations suited to subsequent world modeling and planning. The key challenge is to retain dense visual supervision while endowing this state space with sufficient information, ego-motion sensitivity, and predictive structure.

To instantiate this principle, we propose *MORE*, a V-JEPA framework that directly learns motion-oriented register states. During video pretraining, we introduce a small set of registers into the video encoder and condition them on temporally aligned ego motion. To ensure that these registers retain sufficient visual information, we require masked patch representations to be predicted solely through the registers. Dense patch-level supervision is therefore routed through a compact register bottleneck and directly shapes the state interface used downstream. We further apply blockwise masking and prediction in the register space to explicitly encourage predictive state structure. During joint training, we retain the same register space. A non-autoregressive world model predicts future registers from observed registers and ego motion, while the planner uses both as shared scene memory for trajectory generation and scoring. The world model can therefore model future evolution directly in the state space shaped during pretraining, rather than constructing predictive states from generic visual features. By processing only a small set of registers, MORE also reduces downstream computation, yielding the low inference latency shown in Fig. 1(b) and detailed in Appendix E.

Our contributions are summarized as follows:

- We identify and address the ***state-interface mismatch*** between pretrained visual representations and the states needed for downstream world modeling and planning by learning a shared state interface during video pretraining.
- We introduce motion-oriented registers, register-mediated masked patch prediction, and blockwise register masking to jointly promote information sufficiency, motion sensitivity, and predictive structure in the learned state space.
- We use the same register state space for non-autoregressive world modeling and end-to-end trajectory planning, and evaluate the resulting framework on NAVSIM and NAVSIM-v2, together with closed-loop driving evaluation in HUGSIM.

2 Related Works

Joint-Embedding Predictive Architectures and World Modeling. JEPA (LeCun, 2022) shifts prediction from observation space to learned representation space. I-JEPA (Assran et al., 2023) predicts masked image-region representations from visible context, V-JEPA (Bardes et al., 2024) extends this principle to video, and V-JEPA 2 (Assran et al., 2025) scales both model size and pretraining data. V-JEPA 2.1 (Mur-Labadia et al., 2026) further improves fine-grained spatial and temporal representations through dense predictive loss and deep self-supervision. Recent work explores JEPA for action-conditioned latent world modeling. V-JEPA 2-AC (Assran et al., 2025) freezes a pretrained video encoder and predicts future dense patch features conditioned on actions, while systematic studies of JEPA-WMs examine multi-step latent prediction, action conditioning, and planning strategies (Terver et al., 2025). For autonomous driving, Drive-JEPA (Wang et al., 2026b) transfers V-JEPA representations to trajectory learning. These works demonstrate the potential of JEPA for prediction and control, but the downstream state is generally inherited from visual pretraining, with action-conditioned state evolution introduced mainly during post-training. We instead shape the state space used for future modeling and planning directly during video pretraining.

World Models for Autonomous Driving. Autonomous-driving world models predict the future in observation space, structured spatial representations, or latent feature space. DriveDreamer (Wang et al., 2024a) and Drive-WM (Wang et al., 2024b) generate action- or trajectory-conditioned images and videos; DriveVLA-W0 (Li et al., 2026c) uses future image prediction as dense self-supervision. These approaches preserve rich visual information but also model appearance details not directly relevant to planning. OccWorld (Zheng et al., 2024) and Drive-OccWorld (Yang et al., 2025) instead predict occupancy states, DriveWorld (Min et al., 2024) and DriveX (Shi et al., 2025) model dynamics in structured scene representations, and WoTE (Li et al., 2025c) and ResWorld (Zhang et al., 2026) use future BEV states to evaluate or refine trajectories. Such representations provide explicit geometry but require a full spatial grid at every future step.

Latent world models avoid explicit observation reconstruction by predicting in feature space (Li et al., 2025b; Lu et al., 2026b; Zheng et al., 2025; Lu et al., 2026a). LAW (Li et al., 2025b) uses ego-trajectory-conditioned future latent prediction as an auxiliary objective, World4Drive (Zheng et al., 2025) predicts future visual latents for trajectory selection, and WorldRFT (Yang et al., 2026a) aligns latent representations with planning through geometric and decision supervision. These methods validate latent prediction for driving, but their states are typically derived from existing visual representations and adapted to dynamics and planning through downstream objectives. Dense latent grids also keep state size tied to visual resolution. We instead pretrain the downstream state space itself, using compact registers as both future prediction targets and planning context.

3 Method

We propose *MORE*, which learns motion-oriented registers as a compact state interface for driving world models and planning. As shown in Fig. 2, our pipeline consists of two stages. During pretraining, ego-motion conditioning, register-mediated patch prediction, and blockwise register masking shape an informative and predictable register state space from dense supervision. During joint training, we reuse the same register space to train the world model, trajectory generator, and scorer.

3.1 Preliminaries on Joint-Embedding Predictive Architectures

Joint-Embedding Predictive Architectures (JEPA) (LeCun, 2022) learn representations by predicting latent targets rather than reconstructing observations. In the asymmetric online–target formulation, an online encoder processes a corrupted view, while a target encoder updated by the exponential moving average (EMA) provides targets from the corresponding clean view.

V-JEPA (Bardes et al., 2024) applies this principle to masked video representation prediction. Let $\mathbf{y}$ denote the complete video, $\mathcal{M}$ the set of masked spatiotemporal patch indices, and $\mathbf{x}$ the visible input. The online encoder E_θ processes $\mathbf{x}$, and the predictor P_ϕ uses its outputs together with position-aware mask tokens $\mathbf{\Delta}_\mathcal{M}$ to predict the targets produced by the EMA encoder $E_{\bar{\theta}}$

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left\| P_\phi \left(E_\theta(\mathbf{x}), \mathbf{\Delta}_\mathcal{M} \right)_i - \mathrm{sg} \left(E_{\bar{\theta}}(\mathbf{y})_i \right) \right\|_1 . \tag{1}$$

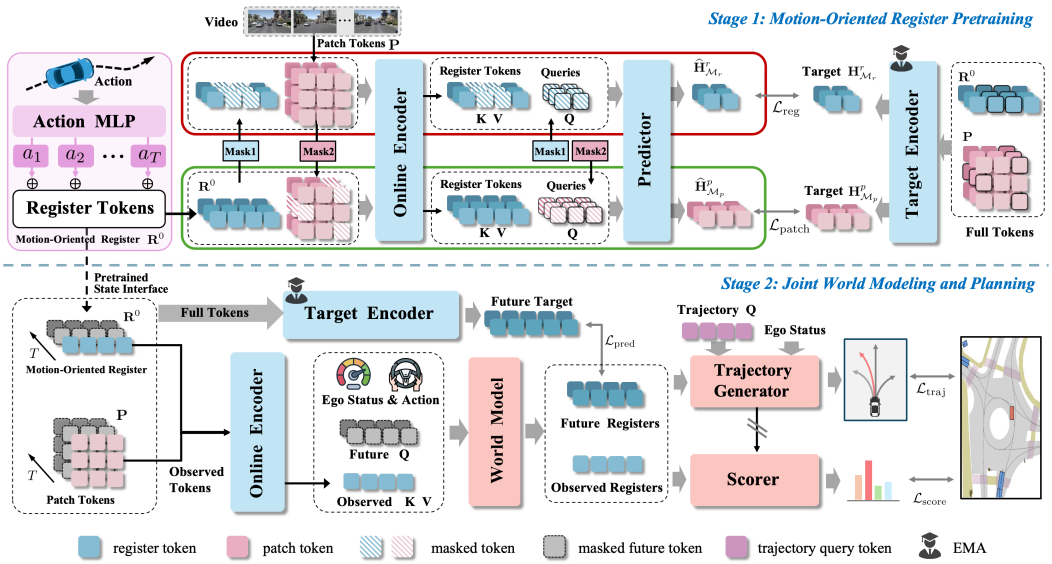


Figure 2: **Overview of *MORE*.** During pre-training, motion-oriented registers are learned through register-mediated masked patch prediction and blockwise register masking. During joint training, the same register space is used for world modeling and as shared scene memory for trajectory generation and scoring.

Here, $\mathrm{sg}(\cdot)$ denotes stop-gradient, and $\bar{\theta}$ is updated by an EMA of θ. V-JEPA provides dense spatiotemporal supervision, but its patch representations are not explicitly shaped as the state interface required for downstream world modeling and planning. We retain these dense targets while pretraining compact register states that are reused for future prediction and planning.

3.2 Motion-Oriented Register Pretraining

Given a video clip $\mathbf{I} \in \mathbb{R}^{T\times H\times W\times 3}$, we tokenize it into non-overlapping tubelets of size $\tau \times p \times p$, yielding a token grid $\mathbf{P} \in \mathbb{R}^{N_t\times N_h\times N_w\times D}$, where $N_t = T/\tau$, $N_h = H/p$, and $N_w = W/p$. For each temporal chunk k, we denote its $N_p = N_h N_w$ patch tokens by $\mathbf{P}_k \in \mathbb{R}^{N_p\times D}$ and introduce $N_r \ll N_p$ learnable registers that interact with them through self-attention.

Patch tokens retain dense visual supervision during pretraining, while only the contextualized registers are reused as the state interface for future modeling and planning. We learn these motion-oriented registers through three complementary designs: ego-motion conditioning, register-mediated masked patch prediction, and blockwise register masking.

Motion-Oriented Register States. We condition the registers on temporally aligned ego-motion. For adjacent frames, we define the relative ego-motion as

$$\mathbf{a}_t = (\Delta x_t, \Delta y_t, \Delta\psi_t)^\top, \qquad t = 1, \dots, T-1, \tag{2}$$

where the motion from frame t to $t+1$ is expressed in the ego coordinate system at frame t, and Δx_t, Δy_t, and $\Delta\psi_t$ denote the two translation components and the heading change. We align these frame-level motions with temporal chunks to obtain $\widetilde{\mathbf{A}} = [\widetilde{\mathbf{a}}_k^\top]_{k=1}^{N_t}$. Under the default setting $\tau = 2$, $\widetilde{\mathbf{a}}_k = \mathbf{a}_{2k-1}$; for larger τ, consecutive motions are composed in $\mathrm{SE}(2)$.

Let $\mathbf{E}^{\mathrm{reg}} \in \mathbb{R}^{N_t\times N_r\times D}$ denote all register embeddings, and let $\mathbf{W}_a \in \mathbb{R}^{3\times D}$ be a learnable action projection matrix. The motion-oriented input to the r-th register in the k-th chunk is defined as

$$\mathbf{R}^0_{k,r} = \mathbf{E}^{\mathrm{reg}}_{k,r} + \widetilde{\mathbf{a}}_k^\top \mathbf{W}_a, \quad k = 1, \dots, N_t, \quad r = 1, \dots, N_r, \tag{3}$$

where $\mathbf{E}^{\mathrm{reg}}_{k,r}$ is the learnable embedding of the r-th register in the k-th chunk. Registers within each chunk share the same motion condition. Injecting aligned ego motion encourages the state space to capture factors related to scene dynamics and state evolution before downstream world modeling.

Two-Branch Pretraining with Shared Encoders. Building on the motion-oriented registers, we pretrain two complementary branches, namely Register-Mediated Masked Patch Prediction and Blockwise Register Masking. Both share the online encoder E_θ, predictor P_ϕ, and EMA target encoder $E_{\bar{\theta}}$. Patch prediction removes a subset of patch tokens while retaining all registers. Register masking retains all patches while removing a subset of registers. Their losses jointly optimize the shared model and constrain a common register state space. We describe each branch below.

Register-Mediated Masked Patch Prediction. Simply introducing registers does not ensure that dense patch-level supervision effectively shapes them, since masked patch prediction can still rely directly on contextualized patch features. We therefore use the contextualized registers as the sole visual context for predicting masked patch representations.

Given the patch grid of size $N_t \times N_h \times N_w$, we sample contiguous spatiotemporal blocks with masked indices $\mathcal{M}_p$ and visible complement $\overline{\mathcal{M}}_p$. The online encoder processes the visible patches together with all action-conditioned registers

$$(\mathbf{P}_c^p, \mathbf{R}_c^p) = E_\theta\Big(\mathbf{P}_{\overline{\mathcal{M}}_p}, \mathbf{R}^0\Big)\,. \tag{4}$$

where $\mathbf{P}_c^p$ and $\mathbf{R}_c^p$ denote the contextualized patch and register outputs, respectively. Self-attention integrates visible patch information into $\mathbf{R}_c^p$. We then discard $\mathbf{P}_c^p$, and predict the masked patch representations using only $\mathbf{R}_c^p$ and position-aware queries

$$\widehat{\mathbf{H}}^p_{\mathcal{M}_p} = P_\phi\Big(\mathbf{Q}^p_{\mathcal{M}_p}, \mathbf{R}_c^p\Big)\,, \tag{5}$$

where $\mathbf{Q}^p_{\mathcal{M}_p}$ follows the same spatiotemporal indices as $\mathcal{M}_p$.

The supervision targets $\mathbf{H}^p_{\mathcal{M}_p}$ are obtained from the EMA target encoder $E_{\bar{\theta}}$, which processes the complete patch sequence with the corresponding action-conditioned registers

$$(\mathbf{H}^p, \mathbf{H}^r) = E_{\bar{\theta}}\big(\mathbf{P}, \mathbf{R}^0\big)\,, \quad \mathbf{H}^p_{\mathcal{M}_p} = \mathrm{sg}(\mathbf{H}^p[\mathcal{M}_p])\,, \tag{6}$$

where $\mathbf{H}^p[\mathcal{M}_p]$ is the target patches at the masked indices. Each masked patch retains a spatially indexed latent target, while all visual context for prediction passes through the register bottleneck. This encourages the compact state to retain the information needed to recover patch latents.

Blockwise Register Masking. The patch-prediction branch encourages registers to retain sufficient visual information but does not directly impose predictive structure on them. We therefore organize the registers as an $N_t \times N_r$ temporal chunk-slot grid and mask contiguous blocks $\mathcal{M}_r$, with $\overline{\mathcal{M}}_r$ denoting the visible complement. The online encoder processes all patches and visible registers

$$(\mathbf{P}_c^r, \mathbf{R}_c^r) = E_\theta\Big(\mathbf{P}, \mathbf{R}^0_{\overline{\mathcal{M}}_r}\Big)\,. \tag{7}$$

We discard contextualized patches $\mathbf{P}_c^r$ and use the shared predictor P_ϕ to recover the masked register states from $\mathbf{R}_c^r$

$$\widehat{\mathbf{H}}^r_{\mathcal{M}_r} = P_\phi\big(\mathbf{Q}^r_{\mathcal{M}_r}, \mathbf{R}_c^r\big)\,, \tag{8}$$

where $\mathbf{Q}^r_{\mathcal{M}_r}$ corresponds one-to-one with the masked registers. The targets are selected from the full register latents $\mathbf{H}^r$ produced by the EMA target encoder in Eq. 6

$$\mathbf{H}^r_{\mathcal{M}_r} = \mathrm{sg}(\mathbf{H}^r[\mathcal{M}_r])\,. \tag{9}$$

By predicting masked registers during pretraining, we establish the register space itself as a prediction target, which the downstream world model directly reuses for future-state prediction, reducing the mismatch between pretraining and world modeling.

3.3 Joint World Modeling and Planning

After representation pretraining, we initialize the online and EMA target encoders with the pretrained weights and retain only the registers as the shared state interface during joint training. Given T_o observed frames from a T-frame video, we define $N_t^o = T_o/\tau$ and $N_t^f = (T - T_o)/\tau$ observed and future chunks, corresponding to $N_o = N_t^o N_r$ and $N_f = N_t^f N_r$ register tokens, respectively.

Future Register World Modeling. The online encoder receives only the observed video and ego-motion to produce the observed registers

$$\mathbf{Z}_o = E_\theta(\mathbf{I}_{1:T_o}, \mathbf{A}_{1:T_o-1}) \in \mathbb{R}^{B\times N_o\times D}. \tag{10}$$

During training, the EMA target encoder processes the full video and its temporally aligned ego-motion. Let $\mathcal{T}_f$ denote the register indices after the observation window. The complete target registers and the corresponding future targets are given by

$$\mathbf{H} = E_{\bar\theta}(\mathbf{I}_{1:T}, \mathbf{A}_{1:T-1}) \in \mathbb{R}^{B\times(N_o+N_f)\times D}, \quad \mathbf{H}_f = \mathrm{sg}(\mathbf{H}[\mathcal{T}_f]) \in \mathbb{R}^{B\times N_f\times D}. \tag{11}$$

The future video and ego-motion are used only to construct training targets and are not provided to the online branch or inference path.

We assign one learnable query per future temporal chunk–register slot pair, yielding $\mathbf{Q}_f \in \mathbb{R}^{N_f\times D}$. To maintain consistency between training and inference, the motion condition $\widetilde{\mathbf{A}}^o$ retains the chunk-level ego-motion within the observation window and zero-pads all future positions. Given the current ego state $\mathbf{S}_o$, the world model F_ψ jointly processes the observed registers and future queries

$$\widehat{\mathbf{Z}}_f = F_\psi\Big(\mathbf{Z}_o, \mathbf{Q}_f, \widetilde{\mathbf{A}}^o, \mathbf{S}_o\Big) \in \mathbb{R}^{B\times N_f\times D}. \tag{12}$$

The outputs at the future-query positions are ordered consistently with $\mathbf{H}_f$. All future queries interact with the observed registers and with one another, allowing the model to predict all future states in a single non-autoregressive pass.

Trajectory Generation and Scoring. We concatenate the observed and predicted future registers into a shared scene memory $\mathbf{Z} \in \mathbb{R}^{B\times(N_o+N_f)\times D}$. Conditioned on $\mathbf{Z}$ and the current ego state $\mathbf{S}_o$, the trajectory generator D_γ maps K learnable trajectory queries $\{\mathbf{q}_k^\tau\}_{k=1}^K$ to candidate trajectories

$$\widehat{\boldsymbol{\tau}}^{(k)} = D_\gamma\left(\mathbf{q}_k^\tau, \mathbf{Z}, \mathbf{S}_o\right), \quad k = 1, \ldots, K. \tag{13}$$

An independent scorer S_ω evaluates the candidate trajectories using the same scene memory

$$\widehat{\mathbf{V}} = S_\omega(\mathrm{sg}(\widehat{\boldsymbol{\tau}}), \mathbf{Z}) \in [0,1]^{B\times K\times C_s}, \tag{14}$$

where $\widehat{\mathbf{V}}$ contains C_s quality scores per candidate, with targets constructed from the NAVSIM/PDM driving criteria (Dauner et al., 2024). Stop-gradient prevents the scoring loss from directly updating the trajectory generator through the predicted trajectories, while preserving gradients through $\mathbf{Z}$. Thus, the same register memory supports future prediction, trajectory generation, and scoring.

3.4 Training Objectives and Inference

Stage 1. We supervise the masked patch and register predictions with ℓ_1 losses

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{patch}} + \mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{M}_p|}\left\|\widehat{\mathbf{H}}^p_{\mathcal{M}_p} - \mathbf{H}^p_{\mathcal{M}_p}\right\|_1 + \frac{1}{|\mathcal{M}_r|}\left\|\widehat{\mathbf{H}}^r_{\mathcal{M}_r} - \mathbf{H}^r_{\mathcal{M}_r}\right\|_1. \tag{15}$$

The online encoder and predictor are optimized by this objective, while the target encoder is updated by EMA as $\bar\theta \leftarrow \mu\bar\theta + (1-\mu)\theta$, where $\mu \in [0,1)$ is the EMA momentum.

Stage 2. Future register prediction is supervised by the EMA targets

$$\mathcal{L}_{\text{pred}} = \frac{1}{N_f}\left\|\widehat{\mathbf{Z}}_f - \mathbf{H}_f\right\|_1. \tag{16}$$

The trajectory generator uses a minimum-over-K loss $\mathcal{L}_{\text{traj}}$, while the scorer is trained with $\mathcal{L}_{\text{score}}$ to match oracle scores derived from the NAVSIM/PDM driving criteria (Dauner et al., 2024). Their detailed definitions are provided in Appendix B.2. The joint objective is

$$\mathcal{L}_{\text{joint}} = \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{score}}\mathcal{L}_{\text{score}}, \tag{17}$$

where all loss weights are set to 1. The online encoder, world model, trajectory generator, and scorer are optimized jointly, while the target encoder is updated only by EMA. The oracle target scores are used only during training.

Inference. At test time, the online encoder produces the observed registers, the world model predicts all future registers, and the planner generates and scores K candidate trajectories from the resulting scene memory. We select the highest-scoring candidate for execution.

Table 1: **Comparison on the NAVSIM-v2 benchmark.** EPDMS* uses the original NAVSIM-v2 evaluator, while EPDMS uses the official implementation with corrected human-behavior filtering.

Method	Venue	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS* ↑	EPDMS↑
End-to-End-based Methods												
TransFuser (Chitta et al., 2023)	TPAMI'22	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7	–
ARTEMIS (Feng et al., 2026)	RAL'25	98.3	95.1	98.6	99.8	87.6	97.4	96.5	98.3	89.1	83.1	–
Hydra-MDP++ (Li et al., 2025a)	arXiv'25	98.8	97.8	99.1	**100.0**	84.0	95.3	70.1	–	**96.8**	84.1	–
DiffusionDrive (Liao et al., 2025)	CVPR'25	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	–	84.5
Drive-JEPA (Wang et al., 2026b)	arXiv'26	98.4	98.6	99.1	99.8	88.4	97.8	97.6	97.9	84.8	87.8	–
SparseDriveV2 (Sun et al., 2026)	ECCV'26	98.1	98.1	99.6	99.8	91.1	97.3	96.9	98.2	78.4	86.7	90.1
DrivoR (Kirby et al., 2026)	CVPR'26	99.3	**99.1**	99.7	99.7	87.6	**98.9**	95.8	97.9	73.4	–	90.3
NTR (Li et al., 2026a)	arXiv'26	**99.4**	99.0	99.7	99.7	88.3	**98.9**	95.5	98.0	76.9	–	90.9
VLA-based Methods												
ReCogDrive (Li et al., 2026d)	ICLR'26	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6	–
WAM-Flow (Xu et al., 2026b)	CVPR'26	98.5	94.5	99.5	99.8	86.9	96.8	97.4	97.6	73.9	84.7	–
DriveVLA-W0 (Li et al., 2026c)	ICLR'26	98.5	**99.1**	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1	–
DriveWorld-VLA (jia et al., 2026)	ICML'26	98.6	**99.1**	99.6	99.8	87.4	97.9	97.0	97.8	78.6	–	86.8
CoWorld-VLA (Huang et al., 2026)	arXiv'26	99.1	97.0	99.6	99.9	87.9	98.5	**97.7**	98.2	86.2	86.2	90.0
World-Model-based Methods												
WoTE (Li et al., 2025c)	ICCV'25	98.5	96.8	98.8	99.8	86.1	97.9	95.5	98.3	82.9	–	87.7
DreamerAD (Yang et al., 2026b)	ECCV'26	98.0	97.2	99.5	99.8	87.8	97.4	97.5	98.3	72.4	–	87.7
DriveLaW (Xia et al., 2026)	CVPR'26	98.7	96.9	99.6	99.8	87.5	98.3	97.6	**98.4**	77.4	–	88.6
EponaV2 (Xu et al., 2026a)	arXiv'26	98.5	97.4	99.5	99.9	87.9	98.1	**97.7**	98.2	77.4	–	88.9
Latent-WAM (Wang et al., 2026a)	arXiv'26	98.1	97.3	99.6	99.8	87.7	97.3	97.6	98.1	87.3	–	89.3
Discrete-WAM (Yao et al., 2026)	arXiv'26	98.5	98.2	99.7	99.8	90.5	97.9	97.2	98.3	78.1	87.0	90.4
Ours	–	98.9	99.0	**99.8**	99.6	**91.7**	98.6	97.1	**98.4**	84.8	–	**92.5**

4 EXPERIMENTS

Datasets, Benchmarks & Metrics. We pretrain on OpenScene (Contributors, 2023), a compact redistribution of nuPlan (Caesar et al., 2021) comprising approximately 120 hours of urban driving at 2 Hz, and jointly train the world model and planner on NAVSIM `navtrain`. We evaluate on the `navtest` splits of NAVSIM v1 and v2 (Dauner et al., 2024; Cao et al., 2025), reporting the Predictive Driver Model Score (PDMS) and Extended Predictive Driver Model Score (EPDMS) under their respective protocols. We further evaluate closed-loop driving in HUGSIM (Zhou et al., 2026), where actions affect subsequent states and observations, reporting Route Completion (RC) and HUGSIM Driving Score (HD-Score). Metric details are provided in Appendices C and D.

Implementation Details. We follow Transfuser (Chitta et al., 2023) in using an input resolution of 256×1024, formed by stacking the front-left, front, and front-right images. The first-stage encoder is a ViT-L initialized from V-JEPA 2 (Assran et al., 2025) weights, with a temporal tubelet size of 2 frames and 8 registers per tubelet. We pretrain for 22,000 steps on OpenScene using AdamW on 32 NVIDIA A100 GPUs with a global batch size of 256, an initial learning rate of 1.5×10^{-4} cosine-decayed to 10^{-6}, and weight decay of 0.04. The target-encoder EMA momentum increases from 0.996 to 0.999. For joint training, only the online and target encoders are initialized from the first stage, while the world model and planner are trained from scratch. The model observes 4 frames and predicts the subsequent 8 frames at 2 Hz. We use $K = 64$ trajectory queries. We train on `navtrain` for 10 epochs using 8 NVIDIA RTX 4090 GPUs with a global batch size of 128 and a base learning rate of 2×10^{-4}, applying a 0.1 multiplier to the online encoder.

4.1 MAIN RESULTS

Results on NAVSIM-v2. Tab. 1 compares MORE with SOTA end-to-end, VLA-based, and world-model-based approaches. MORE achieves the highest EPDMS of 92.5, exceeding DrivoR (Kirby et al., 2026), CoWorld-VLA (Huang et al., 2026), and Discrete-WAM (Yao et al., 2026) by 2.2, 2.5, and 2.1, respectively. It also achieves the best Ego Progress (EP), Driving-Direction Compliance (DDC), and History Comfort (HC), indicating strong driving progress without sacrificing directional compliance or comfort. These results show that compact register states learned during video pretraining can provide an effective shared interface for future modeling and planning.

Results on HUGSIM. We further conduct zero-shot closed-loop evaluation on 436 HUGSIM scenarios spanning four difficulty levels, without benchmark-specific training or fine-tuning. As shown in Tab. 2, MORE achieves the highest average Route Completion (RC) of 51.6 and HD-Score of 39.4. These results show that the compact register states generalize beyond NAVSIM and provide effective predictive context for long-horizon driving decisions.

Table 2: **Photorealistic closed-loop evaluation on HUGSIM.** All results use the corrected evaluator with fixes to trajectory-derived heading computation and related issues.

Method	RC↑					HD-Score↑				
	E	M	H	X	Avg.	E	M	H	X	Avg.
UniAD (Hu et al., 2023)	58.6	41.2	40.4	26.0	40.6	48.7	29.5	27.3	14.3	28.9
VAD (Jiang et al., 2023)	38.7	27.0	25.5	23.0	27.9	24.3	9.9	10.4	8.2	12.3
LTF (Chitta et al., 2023)	68.4	40.7	36.9	25.5	41.4	52.8	24.6	19.8	8.1	24.8
GTRS-Dense (Li et al., 2025d)	64.2	50.0	20.7	22.3	38.0	55.5	39.0	11.7	14.3	28.6
DrivoR (Kirby et al., 2026)	85.9	36.6	34.1	30.4	43.7	77.2	19.3	16.8	13.9	28.1
Latent-WAM (Wang et al., 2026a)	84.2	42.5	30.6	35.5	45.9	72.5	24.0	12.2	18.1	28.9
Ours	88.2	59.9	35.7	25.6	**51.6**	77.4	49.4	21.7	11.1	**39.4**

E: Easy, M: Medium, H: Hard, X: Extreme.

Table 3: **Comparison on the NAVSIM-v1 benchmark.** C and L denote camera and LiDAR input, respectively. †: The results are dependent on perception-based annotation.

Method	Venue	Sensors	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
End-to-End-based Methods								
UniAD† (Hu et al., 2023)	CVPR'23	6C	97.8	91.9	92.9	100.0	78.8	83.4
Transfuser (Chitta et al., 2023)	TPAMI'22	3C&L	97.7	92.8	92.8	100.0	79.2	84.0
Hydra-MDP++ (Li et al., 2025a)	arXiv'25	3C	98.6	98.6	95.1	100.0	85.7	91.0
DiffusionDrive† (Liao et al., 2025)	CVPR'25	3C&L	98.2	96.2	94.7	100.0	82.2	88.1
Drive-JEPA (Wang et al., 2026b)	arXiv'26	1C	98.7	96.2	95.5	100	82.9	89.0
iPad† (Guo et al., 2026)	RAL'26	4C	98.6	98.3	94.9	100.0	88.0	91.7
VLA-based Methods								
AutoVLA (Zhou et al., 2025)	NeurIPS'25	3C	98.4	95.6	98.0	99.9	81.9	89.1
UniVLA (Wang et al., 2026c)	ICLR'26	1C	98.3	93.8	94.2	100.0	80.0	85.6
DriveVLA-W0 (Li et al., 2026c)	ICLR'26	1C	98.7	99.1	95.3	99.3	83.3	90.2
ReCogDrive (Li et al., 2026d)	ICLR'26	3C	97.9	97.3	94.9	100.0	87.3	90.8
ExploreVLA (Sheng et al., 2026)	ECCV'26	1C	98.8	98.4	96.5	99.9	83.5	90.4
SGDrive† (Li et al., 2026b)	CVPR'26	1C	98.6	97.8	96.2	100.0	85.8	91.1
World-Model-based Methods								
LAW (Li et al., 2025b)	ICLR'25	1C	96.4	95.4	88.7	99.9	81.7	84.6
World4Drive (Zheng et al., 2025)	ICCV'25	3C	97.4	94.3	92.8	100.0	79.9	85.1
WoTE† (Li et al., 2025c)	ICCV'25	3C&L	98.5	96.8	94.9	99.9	81.9	88.3
WorldRFT (Yang et al., 2026a)	AAAI'26	3C	97.8	96.8	94.0	100.0	81.7	87.8
PWM (Zhao et al., 2025)	NeurIPS'26	1C	98.6	95.9	95.4	100.0	81.8	88.1
ResWorld† (Zhang et al., 2026)	ICLR'26	3C&L	98.9	96.5	95.6	100.0	83.1	89.0
DriveLaW (Xia et al., 2026)	CVPR'26	1C	99.0	97.1	96.7	100.0	81.3	89.1
EponaV2 (Xu et al., 2026a)	arXiv'26	1C	98.6	97.9	95.7	100.0	84.8	90.4
DriveFuture (Hong et al., 2026)	arXiv'26	6C	98.8	99.1	95.4	100.0	84.2	90.7
Ours	–	3C	98.8	98.4	95.6	100.0	90.7	**93.2**

Results on NAVSIM-v1. As shown in Tab. 3, MORE achieves 93.2 PDMS On NAVSIM-v1 `navtest`, outperforming the strong baselines iPad (Guo et al., 2026), SGDrive (Li et al., 2026b), and DriveFuture (Hong et al., 2026) by 1.5, 2.1, and 2.5, respectively. It also achieves the highest Ego Progress (EP) of 90.7 while remaining competitive in safety and compliance. These results show that compact register states provide effective planning representations across benchmarks.

4.2 Ablation Study

Learning the Register State Interface. Tab. 4 shows that simply introducing registers is insufficient. Initializing them only during joint training (#3) or allowing them to interact with patches during pretraining while retaining patch-based prediction (#4) brings limited gains over the baseline (#1). Using registers as the sole visual context for masked patch prediction (#5 VS. #4) raises EPDMS from 85.6 to 88.5 and PDMS from 86.3 to 89.3. Blockwise register masking (#6) further improves them to 89.1 and 90.2. These results show that two branches provide complementary supervision for learning information-rich and predictable compact states.

Action Conditioning. Injecting ego motion improves both the patch-based baseline (#2 VS. #1) and the register-based model (#8 VS. #6), confirming the effectiveness of motion-conditioned pretraining. Our model benefits more and clearly outperforms its patch-based counterpart (#2), reaching 91.8 EPDMS and 93.0 PDMS. This suggests that the register bottleneck better preserves motion- and planning-relevant information while suppressing redundant visual details.

Table 4: **Ablation Study on NAVSIM `navtest`.** All variants use only the front-camera input.

Case	Configuration	Representation Pre-training				Joint Training		Performance	
		+Reg.	Motion	Patch Ctx.	Reg. Mask	State	Plan. Mem.	EPDMS	PDMS
#1	V-JEPA Baseline	✗	✗	Patch	✗	Patch	Obs.+Pred.	84.9	85.9
#2	V-JEPA + Patch Motion	✗	Patch	Patch	✗	Patch	Obs.+Pred.	89.4	89.2
#3	Joint-Initialized Reg.	✗	✗	Patch	✗	Reg.*	Obs.+Pred.	85.7	86.0
#4	Reg. Interaction Only	✓	✗	Patch	✗	Reg.	Obs.+Pred.	85.6	86.3
#5	Register-Mediated Pred.	✓	✗	Reg.	✗	Reg.	Obs.+Pred.	88.5	89.3
#6	Full w/o Motion	✓	✗	Reg.	✓	Reg.	Obs.+Pred.	89.1	90.2
#7	Full w/o Future Pred.	✓	Reg.	Reg.	✓	Reg.	Obs.	91.1	92.4
#8	**Full (Ours)**	✓	Reg.	Reg.	✓	Reg.	Obs.+Pred.	**91.8**	**93.0**

"+Reg." indicates whether registers are used during pre-training; "Motion" denotes the token type receiving ego-motion conditioning during representation pre-training. "Patch Ctx." denotes the context used for masked patch prediction; "Reg. Mask" denotes the blockwise register masking; and "Plan. Mem." denotes the states provided to the planner. * Registers are introduced and initialized only during joint training.

Future-State Prediction. Finally, #7 removes future-register prediction and supplies the planner with observed registers only, while the full model (#8) uses predicted future registers. The resulting improvement in EPDMS and PDMS confirms that the predicted states capture decision-relevant scene evolution beyond the current observation and provide useful context for trajectory planning.

Number of Registers. Tab. 5 studies the number of registers allocated to each temporal chunk. Using 8 registers achieves the best EPDMS of 91.8, while increasing N_r to 16 or 32 provides no further improvement. We use $N_r = 8$ throughout our experiments.

Table 5: **Register-count ablation.**

Register Num.	4	8	16	32
EPDMS↑	91.4	**91.8**	91.6	91.5

4.3 Visualization

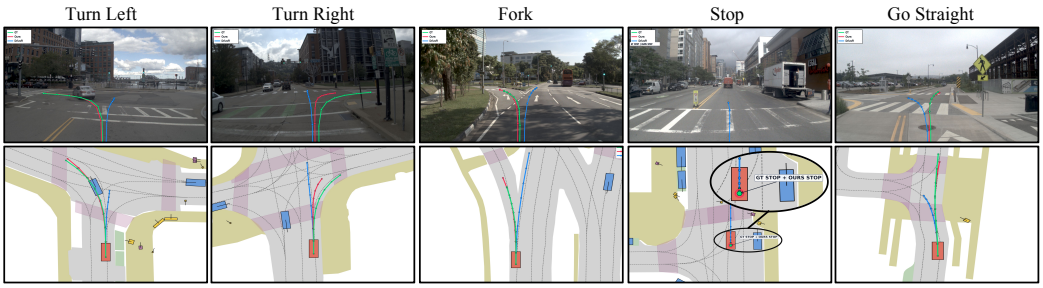


Figure 3: **Qualitative trajectory comparison.** Trajectories: GT, MORE (Ours), DrivoR.

Fig. 3 compares MORE and DrivoR across left and right turns, a fork, stopping, and straight driving. MORE closely follows the ground-truth trajectory across all scenarios, selecting the correct branches, stopping near the reference position, and maintaining route-aligned motion, whereas DrivoR exhibits larger deviations and fails to stop in the stopping scenario. These examples suggest that motion-oriented registers preserve decision-relevant scene dynamics, providing a compact state interface well aligned with downstream planning.

5 Conclusion

We present MORE, which learns motion-oriented registers during video pretraining as a compact state interface for driving world models and planning. During pretraining, ego-motion conditioning, register-mediated masked patch prediction, and blockwise register masking retain dense visual supervision while shaping informative, motion-sensitive, and predictive states. During joint training, the same register space supports non-autoregressive world modeling, trajectory generation, and scoring. Results on NAVSIM-v1, NAVSIM-v2, and HUGSIM demonstrate strong planning performance, low inference latency, and zero-shot closed-loop generalization. Overall, MORE shows that video pretraining can directly shape the compact predictive state required by world modeling and planning rather than leaving this adaptation to downstream training.

AI USE STATEMENT

In this work, we use generative AI tools for research ideation and execution, limited to brainstorming and refining research directions and providing feedback on the research workflow. We also use these tools to improve the clarity, grammar, and organization of author-written text. We didn't use generative AI tools to retrieve or discover literature, draft sections of the manuscript, generate synthetic datasets, or formulate or prove mathematical claims. Other tasks requiring disclosure were not used or are not applicable. The authors critically review and revise all AI-assisted suggestions and language edits, and check technical statements against the implementation and experimental evidence. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

We have made several efforts to ensure the reproducibility of this work. The corresponding sections of the main paper and appendix document the training details and hyperparameters, data splits and preprocessing procedures, evaluation metrics and protocols, and hardware configuration and training cost. We also provide an anonymized code package as supplementary material. Together, these resources document the experimental setup and support reproduction of the reported results.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11618–11628, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

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.

OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven

non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1):226–233, 2026.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(10):11142–11149, 2026.

Yufeng Hong, Xiaotian Zhou, Yingyan Li, Xiangpo Zhou, Lin Liu, Yadan Luo, Shaoqing Xu, Lei Yang, and Ziying Song. Drivefuture: Future-aware latent world models for autonomous driving. *arXiv preprint arXiv:2605.09701*, 2026.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Minqing Huang, Yujiao Xiang, Zihan Liang, Jiajie Huang, Jingqi Wang, Zhi Xu, Feiyang Tan, Hangning Zhou, Mu Yang, and Gong Che. Coworld-vla: Thinking in a multi-expert world model for autonomous driving. *arXiv preprint arXiv:2605.10426*, 2026.

Feiyang jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

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

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2, June 27, 2022.

Jiahui Li, Jiawei Sun, Zixiang Ren, Ming Liu, Jiamin Shi, Ruiteng Zhao, Zhiyang Liu, Liying Liu, Zuoguan Wang, and Kaidi Yang. Ntr: Neural token reconstruction for scene token bottleneck in end-to-end driving. *arXiv preprint arXiv:2605.31116*, 2026a.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4032–4042, June 2026b.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025a.

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*, volume 2025, pp. 42942–42959, 2025b.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146. IEEE, 2025c.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026c.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026d.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025d.

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

Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (3):3292–3310, 2023.

Hongbo Lu, Liang Yao, Chenghao He, Hao Han, Fan Liu, Wenlong Liao, Tao He, and Pai Peng. Rise: Adaptive imagination for world action models. *arXiv preprint arXiv:2608.20430*, 2026a.

Hongbo Lu, Liang Yao, Chenghao He, Haoyu Wang, Xiang Gu, Xianfei Li, Wenlong Liao, Tao He, and Pai Peng. The dawn of world-action interactive models. *arXiv preprint arXiv:2605.11550*, 2026b.

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

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. Explorevla: Dense world modeling and exploration for end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 266–284. Springer, 2026.

Chen Shi, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. Drivex: Omni scene modeling for learning generalizable world knowledge in autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28599–28609. IEEE, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451, 2020.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801. IEEE, 2025.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 446–463. Springer, 2026.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2025.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, et al. Latent-wam: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026a.

Linhan Wang, Zichong Yang, Chen Bai, Guoxiang Zhang, Xiaotong Liu, Xiaoyin Zheng, Xiao-Xiao Long, Chang-Tien Lu, and Cheng Lu. Drive-jepa: Video jepa meets multimodal trajectory distillation for end-to-end driving. *arXiv preprint arXiv:2601.22032*, 2026b.

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*, pp. 55–72. Springer, 2024a.

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

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026c.

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. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pp. 3095–3101. IEEE, 2021.

Jiawei Xu, Zhizhou Zhong, Zhijian Shu, Mingkai Jia, Mingxiao Li, Jia-Wang Bian, Qian Zhang, Kaicheng Zhang, Jin Xie, Jian Yang, et al. Eponav2: Driving world model with comprehensive future reasoning. *arXiv preprint arXiv:2605.14696*, 2026a.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24918–24928, 2026b.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11649–11657, 2026a.

Pengxuan Yang, Yupeng Zheng, Deheng Qian, Zebin Xing, Qichao Zhang, Linbo Wang, Yichen Zhang, Shaoyu Guo, Zhongpu Xia, Qiang Chen, et al. Dreamerad: Efficient reinforcement learning via latent world model for autonomous driving. *arXiv preprint arXiv:2603.24587*, 2026b.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025.

Ziyang Yao, Haochen Liu, Yuncheng Jiang, Zeyu Zhu, Zibin Guo, Jingru Wang, Tianle Liu, Jianwei Cui, Kuiyuan Yang, Hongwei Xie, et al. Discrete-wam: Unified discrete vision-action token editing for world-policy learning. *arXiv preprint arXiv:2606.05645*, 2026.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. Resworld: Temporal residual world model for end-to-end autonomous driving. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 6471–6483, 2026.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 134585–134611. Curran Associates, Inc., 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*, pp. 55–72. Springer, 2024.

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

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4): 4673–4691, 2026.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 27920–27956. Curran Associates, Inc., 2025.

APPENDIX

A Overall **15**

B Datasets and More Implementation Details **15**

B.1 Datasets 15

B.2 Trajectory and Scoring Losses 16

B.3 More Implementation Details 16

C NAVSIM Evaluation **16**

D HUGSIM Closed-Loop Evaluation **17**

E Efficiency analysis **18**

F More Visualization **18**

F.1 Additional Qualitative Trajectory Comparisons 18

F.2 Register Similarity Analysis 18

F.3 Failure Case Analysis 18

A OVERALL

In this appendix, we provide supplementary materials supporting the main paper. Appendix B describes the datasets, trajectory and scoring losses, and further implementation details. Appendices C and D detail the NAVSIM and HUGSIM evaluation protocols, respectively. Appendix E presents the efficiency analysis, while Appendix F provides additional trajectory comparisons, register-similarity analysis, and representative failure cases.

B DATASETS AND MORE IMPLEMENTATION DETAILS

B.1 DATASETS

OpenScene. OpenScene is a redistribution of nuPlan driving logs sampled at 2 Hz, comprising approximately 120 hours of driving. We use its `trainval` logs for motion-oriented register pretraining. Consecutive windows start 1 frame (0.5 s) apart. Each window provides a 12-frame video clip to the model. No OpenScene `test` logs are used for pretraining.

NAVSIM Training and Evaluation Splits. For joint training of the world model and planner, we load NAVSIM's `navtrain`. Frames are sampled at 2 Hz, and consecutive candidate windows start 1 frame (0.5 s) apart. After scene filtering and data-validity checks, the training set contains 103,288 samples. For each sample, the model observes 4 frames and uses the following 8 frames for future-state and trajectory supervision.

For evaluation, the NAVSIM's `navtest` filter selects 12,146 scenarios. We report PDMS under the NAVSIM-v1 protocol and EPDMS under the NAVSIM-v2 protocol on this same `navtest` set.

HUGSIM. HUGSIM is a photorealistic closed-loop simulator in which the ego vehicle's actions affect subsequent states and rendered observations. Its benchmark draws on reconstructed scenes from KITTI-360 (Liao et al., 2023), Waymo Open Dataset (Sun et al., 2020), nuScenes (Caesar et al., 2020), and PandaSet (Xiao et al., 2021). We evaluate the NAVSIM-trained model without HUGSIM-specific training or fine-tuning on 436 scenarios: 80 Easy, 157 Medium, 96 Hard, and

103 Extreme. This evaluation tests the policy under interactive, long-horizon driving rather than the non-reactive NAVSIM scoring protocol.

B.2 Trajectory and Scoring Losses

Each of the K trajectory queries produces a candidate

$$\widehat{\boldsymbol{\tau}}^{(k)} = \left\{ \left(\widehat{\mathbf{p}}_m^{(k)}, \widehat{\psi}_m^{(k)} \right) \right\}_{m=1}^{M}, \qquad \boldsymbol{\tau}^* = \{(\mathbf{p}_m^*, \psi_m^*)\}_{m=1}^{M}, \tag{18}$$

where M is the prediction horizon, $\widehat{\mathbf{p}}_m^{(k)}, \mathbf{p}_m^* \in \mathbb{R}^2$ are the predicted and expert planar positions, and $\widehat{\psi}_m^{(k)}, \psi_m^*$ are their headings at step m. We define $\mathbf{o}(\psi) = (\cos\psi, \sin\psi)^\top$ to represent heading without angular discontinuities. The minimum-over-K trajectory loss is

$$\mathcal{L}_{\text{traj}} = \min_{1 \le k \le K} \frac{1}{M} \sum_{m=1}^{M} \left[\left\| \widehat{\mathbf{p}}_m^{(k)} - \mathbf{p}_m^* \right\|_1 + \lambda_{\text{yaw}} \left\| \mathbf{o}\left(\widehat{\psi}_m^{(k)} \right) - \mathbf{o}(\psi_m^*) \right\|_1 \right], \tag{19}$$

where λ_{yaw} weights the heading term and is set to 1.

For each candidate, the learned scorer predicts $\widehat{\mathbf{V}}^{(k)} = (\widehat{V}_c^{(k)})_{c=1}^{C_s}$, while the NAVSIM/PDM oracle evaluator (Dauner et al., 2024) constructs the corresponding target $\mathbf{V}^{*(k)} = (V_c^{*(k)})_{c=1}^{C_s} \in [0,1]^{C_s}$. Here, C_s is the number of supervised driving criteria. We optimize the scorer using

$$\mathcal{L}_{\text{score}} = \frac{1}{KC_s} \sum_{k=1}^{K} \sum_{c=1}^{C_s} \text{BCE}\left(\widehat{V}_c^{(k)}, V_c^{*(k)} \right), \tag{20}$$

where $\text{BCE}(\widehat{v}, v) = -v \log \widehat{v} - (1-v)\log(1-\widehat{v})$. Together with the world model loss $\mathcal{L}_{\text{pred}}$, the stage 2 objective is

$$\mathcal{L}_{\text{joint}} = \lambda_{\text{pred}} \mathcal{L}_{\text{pred}} + \lambda_{\text{traj}} \mathcal{L}_{\text{traj}} + \lambda_{\text{score}} \mathcal{L}_{\text{score}}, \tag{21}$$

where $\lambda_{\text{pred}} = \lambda_{\text{traj}} = \lambda_{\text{score}} = 1$. The EMA target encoder receives no gradients, and the oracle evaluator is used only to construct training targets.

B.3 More Implementation Details

Input Processing. For each timestamp, we use synchronized front-left, front, and front-right images. Following TransFuser (Chitta et al., 2023), we remove 28 pixels from the top and bottom of each image and crop 416 pixels from both horizontal sides of the two lateral views. The processed views are concatenated in left–front–right order, resized once to 256×1024. We apply no additional image augmentation to preserve the geometry of the stitched view.

Stage1: Motion-Oriented Register Pretraining. We use a ViT-L video encoder with 24 Transformer blocks, a token dimension of 1,024, and 16 attention heads. Using a temporal tubelet size of 2 frames and a spatial patch size of 16×16, each 12-frame clip yields 6 temporal chunks and 48 registers. A three-dimensional ego-motion vector is linearly projected to the encoder dimension and added to the registers of its corresponding chunk before self-attention. The register-mediated masked patch prediction and blockwise register masking branches share a 12-layer Transformer predictor with a hidden dimension of 384 and 12 attention heads.

Stage2: Joint World Modeling and Planning. Four observed frames yield two temporal chunks and 16 observed registers, while the eight future frames correspond to four chunks and 32 learned future queries. The trajectory generator and scorer each contain 4 Transformer layers with a hidden dimension of 256. We configure AdamW for 10 epochs, with the base learning rate 2×10^{-4}, followed by cosine decay to 10^{-6}. The encoder parameter groups use a learning-rate multiplier of 0.1, giving scheduled rates of 2×10^{-5} at the start and 10^{-7} at the end. Training uses bfloat16 precision.

C NAVSIM Evaluation

NAVSIM (Dauner et al., 2024) evaluates predicted ego trajectories through a lightweight simulation-based protocol. The official simulator executes the trajectory using its vehicle model and evaluates it against map, route, and traffic-agent information.

NAVSIM v1. NAVSIM v1 uses non-reactive simulation, in which the ego vehicle executes the predicted trajectory, while surrounding agents follow their recorded future trajectories. The Predictive Driver Model Score (PDMS) comprises five normalized subscores. No At-Fault Collisions (NC) penalizes collisions for which the ego vehicle is considered responsible, with a softer penalty for collisions with static objects. Drivable Area Compliance (DAC) checks whether the ego footprint remains within the drivable area. Ego Progress (EP) measures progress along the route center relative to a safe upper bound estimated by PDM-Closed. Time-to-Collision (TTC) checks whether the projected TTC under constant velocity and heading remains above a safety threshold. Comfort (C) verifies that trajectory acceleration and jerk remain within predefined bounds. The PDMS is

$$\mathrm{PDMS} = \mathrm{NC} \times \mathrm{DAC} \times \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C}}{12}. \tag{22}$$

NC and DAC act as multiplicative safety penalties, while EP, TTC, and C measure route progress, anticipated collision risk, and dynamic feasibility, respectively.

NAVSIM v2. We apply the NAVSIM v2 metric (Cao et al., 2025) to report the Extended PDMS (EPDMS). In addition to the v1 criteria, EPDMS evaluates Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). Let s_m^{A} and s_m^{H} denote the agent and human-reference scores, respectively, for metric m. The official human-reference filter is defined as

$$\phi_m = \begin{cases} 1, & \text{if } s_m^{\mathrm{H}} = 0, \\ s_m^{\mathrm{A}}, & \text{otherwise.} \end{cases} \tag{23}$$

This filter suppresses a metric-specific penalty when the corresponding human trajectory also violates that criterion. The EPDMS is computed as

$$\begin{aligned} \mathrm{EPDMS} = {} & \phi_{\mathrm{NC}} \times \phi_{\mathrm{DAC}} \times \phi_{\mathrm{DDC}} \times \phi_{\mathrm{TLC}} \\ & \times \frac{5\,\phi_{\mathrm{EP}} + 5\,\phi_{\mathrm{TTC}} + 2\,\phi_{\mathrm{LK}} + 2\,\phi_{\mathrm{HC}} + 2\,\phi_{\mathrm{EC}}}{16}. \end{aligned} \tag{24}$$

LK measures sustained deviation from the lane center, HC evaluates consistency with the observed motion history, and EC measures dynamic consistency across consecutive predictions. DDC and TLC penalize wrong-way driving and traffic-light violations.

D HUGSIM Closed-Loop Evaluation

HUGSIM (Zhou et al., 2026) is a photorealistic, interactive closed-loop simulator constructed using 3D Gaussian Splatting (Kerbl et al., 2023). Unlike NAVSIM's per-scene trajectory evaluation, HUGSIM repeatedly queries the policy throughout an episode. The predicted trajectory is executed, the ego vehicle and surrounding actors are updated, and new observations are rendered from the resulting state. Prediction errors therefore affect subsequent inputs and may accumulate over time.

HUGSIM evaluates NC (No Collision), DAC, TTC, and C under its official protocol, while our main comparison reports Route Completion (RC) and HUGSIM Driving Score (HD-Score) for a consistent benchmark-level comparison, as they summarize task completion and overall closed-loop driving quality. RC measures the fraction of the predefined route completed by the ego vehicle. The official implementation computes RC from the furthest reached route-waypoint index, normalized by 0.9 times the total waypoint count and clipped to 1. At evaluated keyframe t, the driving-quality score is

$$\text{HD-Score}_t = \mathrm{NC}_t \times \mathrm{DAC}_t \times \frac{5\,\mathrm{TTC}_t + 2\,\mathrm{C}_t}{7}. \tag{25}$$

For an episode containing N_{eval} evaluated keyframes, the final score is

$$\text{HD-Score} = \mathrm{RC} \times \frac{1}{N_{\mathrm{eval}}} \sum_{t=1}^{N_{\mathrm{eval}}} \text{HD-Score}_t. \tag{26}$$

HD-Score therefore scales the average closed-loop driving quality by route completion, preventing a stationary but collision-free policy from receiving a high score.

Table 6: **Efficiency.** We compare model parameters, peak GPU memory, and inference latency with respect to NAVSIM-v2 EPDMS.

Method	Img. Size	Obs. Frames	Cams	Params. ↓	Peak Mem. ↓	Latency ↓	EPDMS↑
Epona	512×1024	10	1	2,915.4 M	14.70 GB	900.0 ms	85.1
World4Drive	384×640	4	6	70.2 M	1.98 GB	414.5 ms	84.8
DriveVLA-W0 †	Pre-tokenized	–	–	9,073.3 M	17.23 GB	1,633.5 ms	86.1
DriveLaW	768×1344	4	1	3,301.6 M	6.89 GB	1,167.0 ms	88.6
CoWorld-VLA	512×1024	1	1	5,372.9 M	10.36 GB	378.4 ms	90.0
Drive-JEPA	256×512	2	1	311.2 M	1.83 GB	73.3 ms	87.8
MORE (Ours)	256×1024	4	3	642.7 M	6.18 GB	110.0 ms	**92.5**

E Efficiency Analysis

As shown in Tab. 6, MORE achieves the highest EPDMS of 92.5 while maintaining an inference latency close to Drive-JEPA (Wang et al., 2026b), which does not use an explicit future-state world model (110.0 VS. 73.3 ms). With comparable peak memory (6.18 vs. 6.89 GB), MORE requires only about 1/10 the inference latency of DriveLaW (Xia et al., 2026) (110.0 vs. 1,167.0 ms). These results show that motion-oriented registers provide an effective and efficient shared state interface for future world modeling and planning.

All measurements use one NVIDIA RTX 4090 with batch size 1, serial execution, warm-up, and synchronized CUDA events. Latency is the per-sample median, and memory is peak allocated GPU memory, including resident model weights. Disk I/O, dataset parsing, and PDM simulation are excluded. † DriveVLA-W0 (Li et al., 2026c) is evaluated from its released pre-tokenized multimodal interface. Thus, the number of raw camera views is unspecified, and image tokenization is excluded from the measured latency.

F More Visualization

F.1 Additional Qualitative Trajectory Comparisons

Fig. 4 presents additional comparisons across diverse straight and turning scenarios. Each example shows the front-camera view (top) and BEV visualization (bottom). MORE closely follows the GT travel direction and curvature, including at complex intersections with surrounding traffic. In contrast, DrivoR exhibits larger lateral deviations or continues straight in several turning cases. Together with Fig. 3, these results indicate that motion-oriented registers retain planning-relevant motion and scene context for accurate trajectory generation.

F.2 Register Similarity Analysis

To examine how our pretraining design shapes the register space beyond merely introducing register tokens, we compare the pretrained representations of configuration #4 in Tab. 4 and MORE. As shown in Fig. 5, configuration #4 exhibits broadly high similarity within and across temporal chunks, with a mean of 0.793, indicating substantial redundancy among registers. MORE reduces the mean similarity to 0.653 and produces more differentiated patterns across register slots and temporal chunks. These results show that our register pretraining design encourages complementary rather than highly overlapping representations, yielding a more expressive compact state interface for future world modeling and planning.

F.3 Failure Case Analysis

Fig. 6 presents representative failure cases. In the two near-stop scenarios, the GT remains within approximately 0.85 m of its initial position, whereas MORE advances by 9.9–11.7 m. In the left example, the school bus severely occludes the road and potential cross traffic, and in the right example, the pedestrian crossing and surrounding agents provide ambiguous cues about the required stop

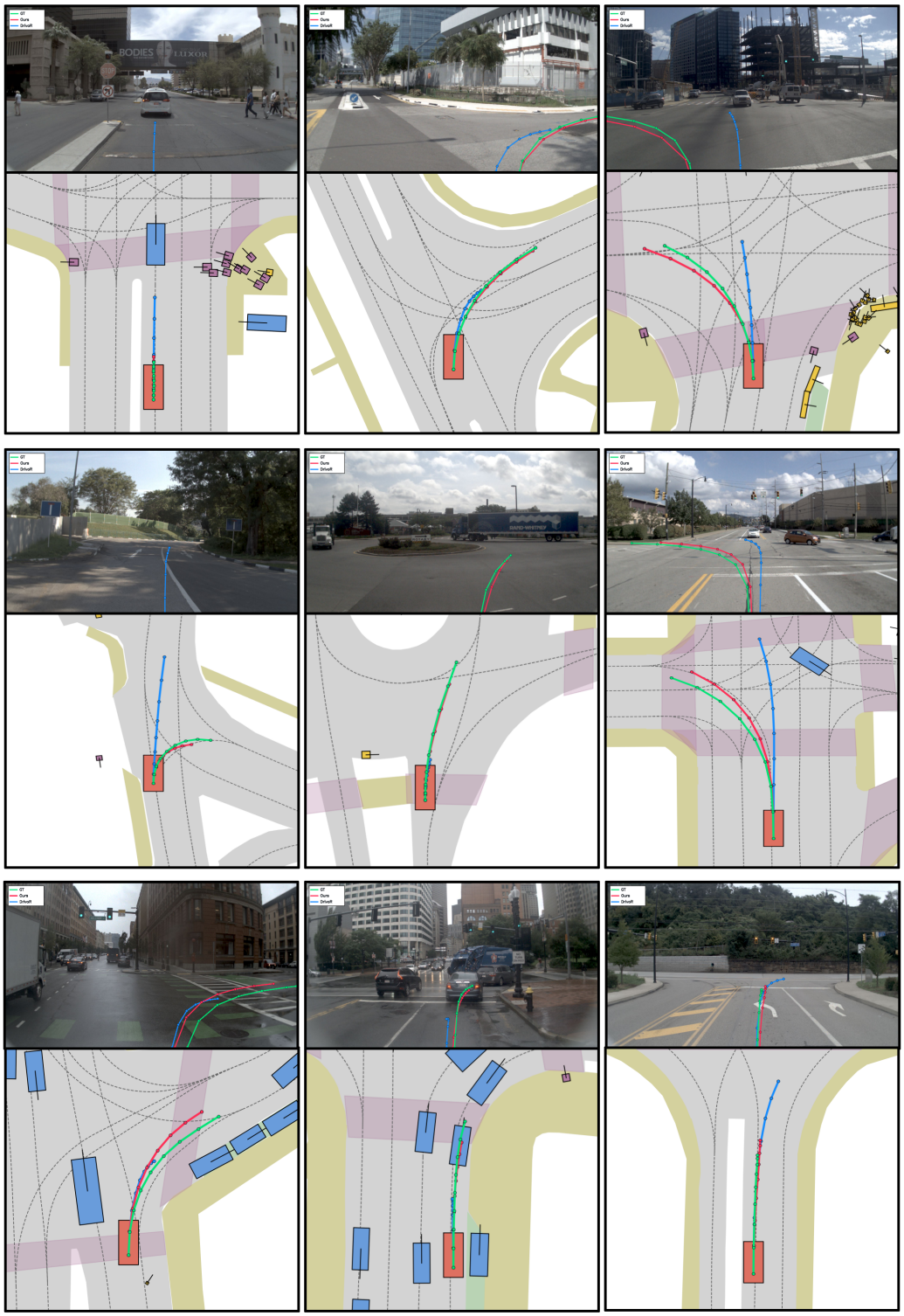
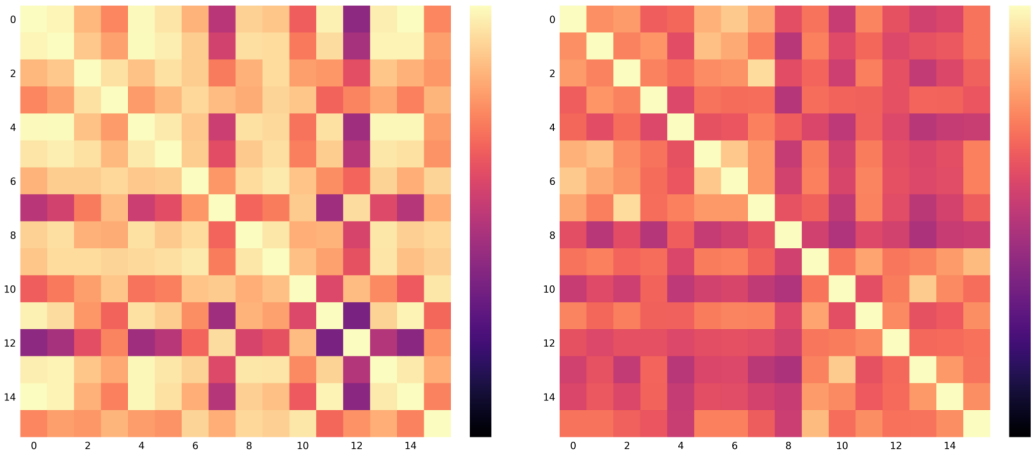


Figure 4: **Additional qualitative comparison.** Trajectories: GT, MORE (Ours), DrivoR.

duration. These errors are consistent with a common challenge in vision-based planning: under partial observability, a short observation window may admit multiple plausible future behaviors. The center example is substantially milder, with an average displacement error (ADE) of 0.46 m and a

final displacement error (FDE) of 1.80 m, and occurs on an unmarked road with branch-like geometry and weak lane guidance. Overall, the failures are concentrated in scenes where the duration of a stop or the intended route continuation is difficult to infer from the available observations. Longer temporal context and uncertainty-aware future prediction may improve robustness in such cases.



(a) Register Interaction Only **(b) Ours**

Figure 5: **Pairwise similarity of pretrained registers.** Register similarities within and across temporal chunks are shown for **(a) configuration #4 in Tab. 4, which enables register–patch interaction while retaining patch-based prediction, and (b) MORE.** Their mean similarities are 0.793 and 0.653, respectively.



Figure 6: **Representative failure cases.** Trajectories: GT, MORE (Ours).