AUTO-JEPA: A LATENT WORLD MODEL OF CONTINUOUS INTENT FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing autonomous-driving world models typically perform dense prediction of future videos, occupancy states, BEV representations, or agent motion. We argue that planning need not reconstruct the complete future world, but only focus on scene features that affect future ego action. Based on this perspective, we propose **Auto-JEPA**, an action-oriented latent world model that learns continuous future driving intent through joint-embedding prediction. Given visual observations, ego-motion history, and navigation commands, Auto-JEPA predicts an intent embedding aligned with the latent representation of the future ego trajectory. The predicted intent retrieves executable trajectories from a fixed trajectory memory, which are then ranked by a scene-conditioned candidate selection module. Auto-JEPA keeps the visual encoder frozen, requires no explicit perception annotations, and uses no learned trajectory generator. By optimizing only task-specific modules for trajectory representation, intent prediction, and candidate selection, Auto-JEPA achieves **91.3 PDMS** on NAVSIM v1 and **89.1 EPDMS** on NAVSIM v2. Semantic occlusion experiments show that masking dynamic-agent regions induces an average intent change $2.97\times$ that of equal-area random masking. Moreover, occluding vehicles that affect future driving substantially changes the predicted intent and selected trajectory, whereas both remain essentially unchanged when non-influential vehicles are occluded. These results show that future-intent prediction encourages the model to focus on planning-relevant visual features and supports high-quality planning without dense future-world modeling.

1 INTRODUCTION

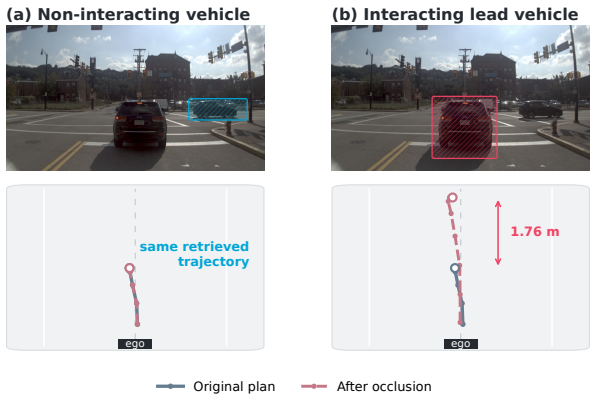


Figure 1: Selective response to action-relevant scene information. Occluding a non-interacting vehicle causes little change in the predicted intent and plan, whereas occluding the interacting lead vehicle shifts the intent and selected trajectory. Annotations are used only for analysis.

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.

World models offer a compelling route toward autonomous driving systems that can reason about how a scene may evolve before committing to an action. Many driving world models, however, formulate this objective as dense future prediction, generating future multiview observations, occupancy fields, or latent states that describe the evolution of the surrounding scene (Wang et al., 2023b; Zheng et al., 2023; Zhang et al., 2026; Jia et al., 2026). A parallel line of work explicitly forecasts the future trajectories and interactions of multiple traffic participants (Seff et al., 2023). Although these objectives preserve rich information about the environment, they require the model to allocate substantial capacity to scene elements whose dynamics may have little influence on the ego vehicle's immediate decision. Predicting all observable entities without regard to their planning relevance can therefore increase computation and propagate perception and forecasting errors into downstream planning.

We argue that a planning-oriented predictive model need not reconstruct the complete future state of a scene, but only focus on scene features that affect future ego action. Based on this perspective, we introduce **Auto-JEPA**, an action-oriented latent world model that learns continuous future driving intent through joint-embedding prediction. Given visual observations, ego-motion history, and a navigation command, it predicts a future ego-trajectory latent whose temporal tokens encode motion geometry and dynamics. Recent work explores planning-aligned latent prediction (Li et al., 2025b; Zheng et al., 2025; Wang et al., 2026; Xie et al., 2026), primarily for representation pretraining, auxiliary supervision, or action distillation. Auto-JEPA uses future ego-trajectory latents for trajectory retrieval.

Auto-JEPA first trains a trajectory encoder and then freezes it to define the future-trajectory target space. Following joint-embedding predictive learning (Assran et al., 2023; Bardes et al., 2024), a visual predictor infers the corresponding future-trajectory latent from the current scene context, ego-motion history, and route command. Latent alignment and contrastive objectives train the predictor without future-image reconstruction. Because its prediction target is defined solely by future ego motion, the model need not preserve all observable scene content and is encouraged to prioritize planning-relevant visual features. At inference, the predicted intent directly serves as the retrieval key of a fixed trajectory memory, and a scene-conditioned module selects the final executable candidate.

The trajectory memory contains recorded, kinematically plausible trajectory geometries encoded by the same trajectory encoder. A scene-conditioned scorer ranks the retrieved candidates, and a learned drivable-area gate screens candidates likely to violate drivable-area constraints before final selection. This design separates complementary responsibilities: the latent predictor determines *what kind of future motion is appropriate*, retrieval provides explicit trajectory geometry, the scorer estimates scene-level driving quality, and the gate reduces drivable-area violations. Although these components are optimized in stages, Auto-JEPA preserves an end-to-end sensor-to-trajectory planning interface without intermediate perception outputs or a learned trajectory generator.

We evaluate Auto-JEPA on NAVSIM v1 and NAVSIM v2 (Dauner et al., 2024). Under this lightweight setting, Auto-JEPA achieves **91.3 PDMS** on NAVSIM v1 and **89.1 EPDMS** on NAVSIM v2. Figure 1 illustrates the different effects of individual vehicles within the same scene: occluding the interacting lead vehicle substantially changes the predicted intent and selected trajectory, whereas both remain essentially unchanged when the non-interacting adjacent vehicle is occluded. To quantify sensitivity to traffic-participant information at the dataset level, we further conduct controlled semantic occlusions on the full validation split. Masking dynamic-agent regions induces an average intent change $2.97\times$ that of equal-area random masking and a larger change in 71.1% of scenes.

Our main contributions are summarized as follows:

- We formulate planning-oriented latent world modeling as future ego-trajectory latent prediction. Using continuous driving intent as the prediction target encourages the model to focus on visual features relevant to ego action, thereby avoiding dense future-scene reconstruction.
- We propose Auto-JEPA, which maps visual context into a future-trajectory latent space through joint-embedding prediction. The predicted intent directly retrieves executable candidates from a fixed trajectory memory, followed by scene-conditioned selection.

- Auto-JEPA achieves **91.3 PDMS** on full NAVSIM v1 navtest and **89.1 EPDMS** on NAVSIM v2; controlled semantic occlusion experiments further show that the model selectively focuses on more critical scene features.

2 Related Work

World Models for Autonomous Driving. Driving world models predict scene evolution in pixel or structured spaces for simulation, representation learning, and planning. Generative approaches synthesize controllable future driving videos from observations, actions, or language (Hu et al., 2023; Wang et al., 2023a;b; Gao et al., 2024), while structured alternatives forecast future occupancy or point clouds (Zheng et al., 2023; Yang et al., 2023). Although these representations capture rich appearance and geometry, they require dense prediction over broad scene content, including details that need not determine the immediate ego plan.

Latent world models reduce explicit reconstruction by forecasting compact future features. LAW, World4Drive, DeepSight, and DriveWorld-VLA predict future scene features, BEV states, or action-conditioned scene evolution for downstream planning (Li et al., 2025b; Zheng et al., 2025; Zhang et al., 2026; Jia et al., 2026). Their targets nevertheless primarily describe how the surrounding scene evolves. Auto-JEPA instead predicts the representation of future ego motion itself, retaining scene dynamics through their implications for the planned trajectory.

Joint-Embedding Predictive Learning. Joint-embedding predictive architectures learn by predicting target representations rather than reconstructing observations. I-JEPA establishes this principle for images (Assran et al., 2023), V-JEPA extends it to video (Bardes et al., 2024), and V-JEPA 2 further demonstrates that self-supervised video representations can support understanding, prediction, and planning (Assran et al., 2025). Drive-JEPA adapts this family to driving-video pretraining and trajectory distillation (Wang et al., 2026). Auto-JEPA differs in the operational role of prediction: the future ego-trajectory latent directly serves as the planner's retrieval key and therefore participates in trajectory planning at inference rather than only supporting training-time representation learning.

End-to-End Trajectory Planning. End-to-end driving maps sensor observations and navigation context directly to an ego trajectory, but a single logged future provides limited coverage of multiple feasible maneuvers. Existing methods score candidates from offline trajectory vocabularies (Jiang et al., 2024; Li et al., 2024; Sun et al., 2026), generate or refine proposals (Liao et al., 2025; Xing et al., 2025; Guo et al., 2025), or combine a learned generator with a trajectory scorer (Ang et al., 2026). Latent trajectory nearest-neighbor search has also been explored for motion forecasting rather than ego planning (Biktairov et al., 2020).

Recent VLA planners tokenize or decode ego actions from visual and language context (Zhou et al., 2025b;a; Li et al., 2026a). They learn parametric action decoders, whereas Auto-JEPA predicts a continuous future-ego-motion intent latent for non-parametric trajectory retrieval.

3 Method

Overview. Auto-JEPA predicts a continuous future ego-motion representation and uses it to retrieve executable trajectories. Given four front-camera frames, four historical ego positions, and a route command, a frozen V-JEPA 2 encoder (Assran et al., 2025) extracts visual tokens, and a Transformer predictor fuses visual, motion, and route context to produce eight temporal latent tokens representing one continuous driving intent. Figure 3 presents the complete training and inference pipeline.

During training, the predicted intent is aligned with the representation of the ground-truth future trajectory produced by a frozen trajectory encoder. At inference, it retrieves the 300 most similar trajectories from a non-parametric memory; a scene-conditioned scorer ranks these candidates and a learned drivable-area gate filters infeasible proposals. This separates intent prediction, trajectory instantiation, quality ranking, and feasibility filtering without reconstructing future observations or directly regressing waypoint coordinates.

3.1 Future Ego-Motion Intent Representation

Figure 2 details the two-stage learning process: trajectory-space pretraining followed by visual intent prediction. For each driving scene, we represent the ground-truth future ego trajectory as eight two-dimensional waypoints, $\mathbf{Y} = [(x_1, y_1), \ldots, (x_8, y_8)] \in \mathbf{R}^{8\times 2}$.

Before training the visual intent predictor, we learn the target space with a trajectory autoencoder composed of a trajectory encoder E_{traj} and a lightweight decoder D_{traj}. The encoder maps a future trajectory to a sequence of latent tokens, and the decoder reconstructs the trajectory from this representation, $\mathbf{Z}^+ = E_{\text{traj}}(\mathbf{Y}) \in \mathbf{R}^{8\times 1024}$, $\hat{\mathbf{Y}} = D_{\text{traj}}(\mathbf{Z}^+)$. The autoencoder is optimized using a trajectory reconstruction objective, $\mathcal{L}_{\text{traj}} = \mathcal{L}_{\text{xy}} + \lambda_e \mathcal{L}_{\text{end}} + \lambda_v \mathcal{L}_{\text{vel}} + \lambda_a \mathcal{L}_{\text{acc}}$, where the four terms supervise waypoint coordinates, the final endpoint, velocity, and acceleration, respectively. After this pretraining stage, the decoder is discarded and E_{traj} is frozen. The frozen encoder then defines the target latent $\mathbf{Z}^+$ for intent prediction and encodes every trajectory in the retrieval memory, ensuring that prediction targets and retrieval candidates occupy the same latent space. The eight latent tokens preserve trajectory time, geometry, and motion and jointly describe one continuous future realization rather than eight maneuver classes. We refer to this representation as the *driving intent*. Using the same encoder for supervision and memory construction places predicted intents and executable trajectories in a shared space, allowing the intent latent to serve directly as the retrieval key.

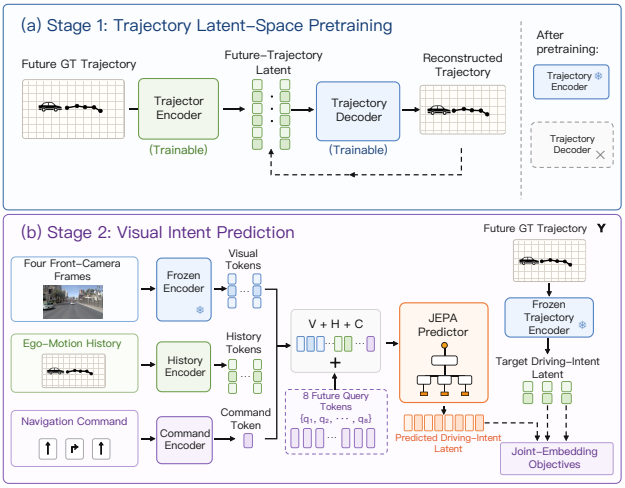


Figure 2: Trajectory-space pretraining and visual intent prediction in Auto-JEPA. Stage 1 learns the future-trajectory target space with a trajectory autoencoder; the decoder is then discarded and the trajectory encoder is frozen. Stage 2 predicts a continuous future driving-intent latent from visual observations, ego-motion history, and navigation commands, and aligns it with the frozen trajectory target representation.

3.2 Visual Intent Prediction

Given the current visual observation, ego-motion history, and navigation command, Auto-JEPA predicts the corresponding future ego-motion latent. We denote the model input as $\mathbf{X} = (\mathbf{I}, \mathbf{H}, \mathbf{C})$, where $\mathbf{I}$ contains four historical frames from the front-facing camera, $\mathbf{H} \in \mathbf{R}^{4\times 2}$ contains four historical ego positions, and $\mathbf{C} \in \mathbf{R}^4$ denotes the route command. A frozen V-JEPA 2 visual encoder E_{vis} extracts visual tokens, $\mathbf{F}_v = E_{\text{vis}}(\mathbf{I})$. The history encoder E_{hist} and command encoder E_{cmd} map the remaining inputs to the predictor feature dimension, $\mathbf{F}_h = E_{\text{hist}}(\mathbf{H})$, $\mathbf{F}_c = E_{\text{cmd}}(\mathbf{C})$. The visual, history, and command features condition a JEPA predictor P_θ composed of 24 Transformer blocks (Vaswani et al., 2017) with a hidden dimension of 1024 and 16 attention heads. The predictor outputs eight future latent tokens, $\hat{\mathbf{Z}} = P_\theta(\mathbf{F}_v, \mathbf{F}_h, \mathbf{F}_c) \in \mathbf{R}^{8\times 1024}$. The prediction $\hat{\mathbf{Z}}$ matches the temporal organization and feature dimension of $\mathbf{Z}^+$. During driving-domain training, the visual encoder remains frozen, while the history encoder, command encoder, and JEPA predic-

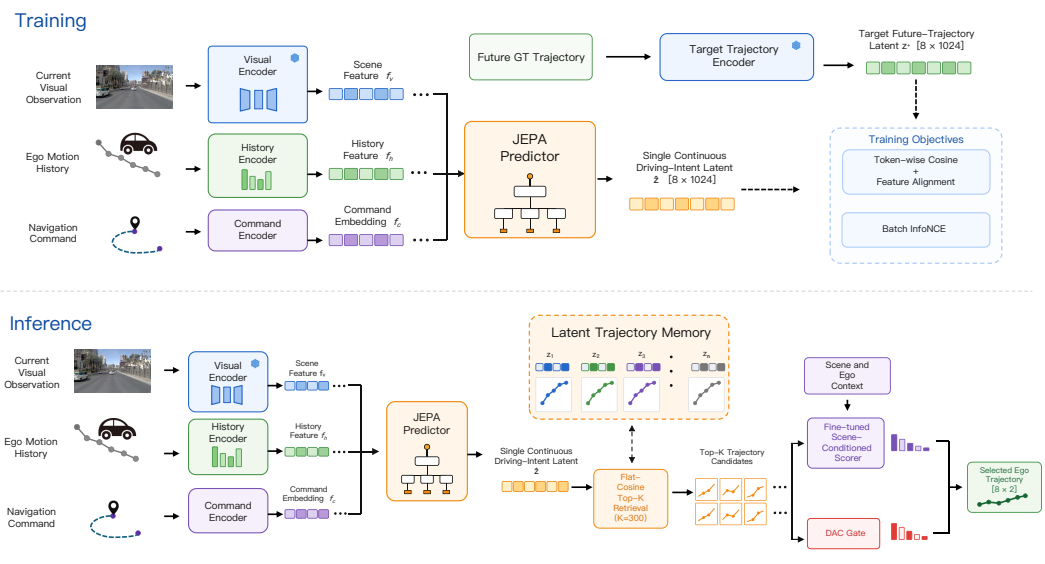


Figure 3: Overview of Auto-JEPA. During training, the predictor learns a continuous future ego-motion intent by aligning its predicted latent with the representation of the ground-truth future trajectory. During inference, the predicted intent retrieves 300 candidates from a ground-truth-only latent trajectory memory; a scene-conditioned scorer ranks candidate quality and an independent feasibility gate filters drivable-area violations before final selection. Snowflake symbols indicate frozen modules.

tor are optimized to aggregate visual evidence, ego dynamics, and navigation constraints without explicitly predicting intermediate scene states.

3.3 Joint-Embedding Training Objectives

Rather than directly supervising waypoint coordinates with ADE or FDE, Auto-JEPA optimizes predictions in the frozen trajectory latent space using feature alignment, token-wise cosine alignment, and batch-level InfoNCE (van den Oord et al., 2018).

Feature alignment. We first normalize the predicted and target latents and apply a Smooth L1 loss to their feature values, $\mathcal{L}_{\text{feat}} = \text{SmoothL1}\Big(\text{Norm}(\hat{\mathbf{Z}}), \text{Norm}(\mathbf{Z}^+)\Big)$.

Token-wise cosine alignment. For the eight future time positions, we minimize the cosine distance between corresponding predicted and target tokens,

$$\mathcal{L}_{\cos} = \frac{1}{8}\sum_{t=1}^{8}\left(1 - \frac{\hat{\mathbf{z}}_t^\top \mathbf{z}_t^+}{\|\hat{\mathbf{z}}_t\|_2 \|\mathbf{z}_t^+\|_2}\right). \tag{1}$$

Batch-level InfoNCE. Positive alignment alone may map distinct driving scenes to similar representations. We therefore flatten and normalize each complete latent sequence, $\hat{\mathbf{q}}_i = \text{Norm}(\text{vec}(\hat{\mathbf{Z}}_i))$, $\mathbf{k}_j = \text{Norm}(\text{vec}(\mathbf{Z}_j^+))$. For scene i, its target latent is the positive and target latents from other scenes serve as negatives,

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\hat{\mathbf{q}}_i^\top \mathbf{k}_i/\tau)}{\sum_{j=1}^{B}\exp(\hat{\mathbf{q}}_i^\top \mathbf{k}_j/\tau)}, \tag{2}$$

where $\tau = 0.07$. During distributed training, target latents are gathered across GPUs to enlarge the negative set.

The complete intent-prediction objective is $\mathcal{L}_{\text{intent}} = 0.1\mathcal{L}_{\text{feat}} + 2.0\mathcal{L}_{\cos} + \mathcal{L}_{\text{NCE}}$. Together, these objectives align local features and temporal-token semantics while preserving discrimination across driving scenes, making the predicted latent suitable for trajectory retrieval.

3.4 NON-PARAMETRIC TRAJECTORY RETRIEVAL

To ground the predicted continuous intent into explicit trajectory geometry, we construct a non-parametric trajectory memory using only ground-truth driving trajectories. Each memory trajectory $\mathbf{Y}_n$ is encoded by the same frozen trajectory encoder used to define the training targets, $\mathbf{Z}_n = E_{\text{traj}}(\mathbf{Y}_n) \in \mathbf{R}^{8\times 1024}$. The resulting memory is $\mathcal{M} = \{(\mathbf{Z}_n, \mathbf{Y}_n)\}_{n=1}^{N}, \quad N = 110{,}335$. Each entry retains its latent and waypoint coordinates, directly providing candidate geometry. For a predicted intent $\hat{\mathbf{Z}}$, we flatten and L2-normalize the query and memory latents, $\mathbf{q} = \text{Norm}(\text{vec}(\hat{\mathbf{Z}})), \quad \mathbf{m}_n = \text{Norm}(\text{vec}(\mathbf{Z}_n))$. Their retrieval similarity is the flat cosine similarity $r_n = \mathbf{q}^\top \mathbf{m}_n$. We rank the complete memory by r_n and retrieve the $K = 300$ most similar trajectories, $\mathcal{C} = \text{TopK}\big(\{r_n\}_{n=1}^{N}, K\big)$. Retrieval identifies intent-compatible geometry, while subsequent modules handle scene-level safety. Because candidates come from recorded trajectories, inference requires neither an additional learned trajectory generator nor iterative waypoint sampling.

3.5 SCENE SCORING AND FEASIBILITY GATING

Latent similarity measures intent compatibility but not scene-level safety. We therefore use a scene-conditioned quality scorer and an independently trained drivable-area feasibility gate.

Scene-conditioned utility branch. Given scene features $\mathbf{F}_{\text{scene}}$, ego context $\mathbf{e}$, and a candidate trajectory $\mathbf{Y}_k$, the scorer predicts a quality score $s_k = S_\phi(\mathbf{F}_{\text{scene}}, \mathbf{e}, \mathbf{Y}_k)$.

We initialize S_ϕ from the publicly released CLOVER trajectory scorer (Ang et al., 2026) and re-optimize its trainable modules on the ground-truth-only candidates retrieved by Auto-JEPA. Training uses collision, drivable-area, time-to-collision, comfort, and ego-progress supervision (Dauner et al., 2024), together with a within-scene ranking objective. The scorer and gate are trained exclusively on candidates from the NAVSIM training split, using labels generated offline by the NAVSIM/CLOVER `get_sub_score` evaluator with the batched `navsim_v1_style` relabeling protocol. This protocol disables per-proposal two-way rollout during label generation; the final benchmark results are evaluated separately under the official NAVSIM protocols. Scorer training is separate from intent prediction, with no gradients propagated to the visual encoder, JEPA predictor, or trajectory memory.

Drivable-area feasibility gate. To explicitly reject candidates at risk of leaving the drivable region, the independently trained feasibility gate predicts $p_k^{\text{DAC}} = G_\psi(\mathbf{F}_{\text{scene}}, \mathbf{e}, \mathbf{Y}_k)$, where p_k^{DAC} is the predicted probability of a DAC failure. At inference, we form a safety mask using $\tau_{\text{DAC}} = 0.2$, $m_k = \mathbf{1}\big[p_k^{\text{DAC}} \le \tau_{\text{DAC}}\big]$. Only candidates that pass the gate participate in final selection. If all are rejected, the system falls back to ungated utility ranking. Evaluator labels are unavailable to the gate at inference time.

The final trajectory is selected by a masked argmax over utility scores, $k^* = \arg\max_{k:m_k=1} s_k, \quad \mathbf{Y}^* = \mathbf{Y}_{k^*}$. Thus, intent prediction supplies the retrieval query, memory provides trajectory geometry, and the scorer–gate cascade performs scene-conditioned selection.

4 EXPERIMENTS

We evaluate the complete retrieval-based planner under the official NAVSIM v1 and v2 protocols. Unless otherwise stated, all experiments use the same intent predictor, ground-truth-only trajectory memory, scene-conditioned scorer, and feasibility gate. We first report benchmark results, then isolate the effects of intent prediction and candidate selection, followed by analyses of candidate-pool size and visual dependence.

4.1 DATASET AND METRICS

NAVSIM v1. NAVSIM is a data-driven, non-reactive benchmark for autonomous-driving planning that provides large-scale trainval data and simulation-based evaluation on navtest (Dauner et al., 2024). The v1 benchmark evaluates each predicted ego trajectory through a simulator and reports no-at-fault collision (NC), drivable-area compliance (DAC), time to collision (TTC), comfort (C), and ego progress (EP). These terms are aggregated into the Predictive Driver Model Score (PDMS):

Table 1: NAVSIM v1 navtest comparison. C/L denotes camera/LiDAR input. Baseline references follow row order: (Chitta et al., 2023; Weng et al., 2024; Li et al., 2024; Liao et al., 2025; Li et al., 2025b; Chen et al., 2025; Li et al., 2025c; Zhang et al., 2025; Zhou et al., 2025b; Li et al., 2026b; Luo et al., 2025; Li et al., 2026a; Chen et al., 2026).

Method	Sensors	NC ↑	DAC ↑	TTC ↑	C ↑	EP ↑	PDMS ↑
Human	–	100.0	100.0	100.0	99.9	87.5	94.8
End-to-End Planning Methods							
TransFuser	3×C+L	97.7	92.8	92.8	**100.0**	79.2	84.0
PARA-Drive	6×C	97.9	92.4	93.0	99.8	79.3	84.0
Hydra-MDP	3×C+L	98.3	96.0	94.6	**100.0**	78.7	86.5
DiffusionDrive	3×C+L	98.2	96.2	94.7	**100.0**	82.2	88.1
World-Model-Based Methods							
LAW	1×C	96.4	95.4	88.7	99.9	81.7	84.6
DrivingGPT	1×C	98.9	90.7	94.9	95.6	79.7	82.4
WoTE	3×C+L	98.5	96.8	94.4	99.9	81.9	88.3
Epona	3×C	97.9	95.1	93.8	99.9	80.4	86.2
VLA-Based Methods							
AutoVLA	3×C	98.4	95.6	**98.0**	99.9	81.9	89.1
ReCogDrive	3×C	98.2	97.8	95.2	99.8	83.5	89.6
AdaThinkDrive	1×C	98.4	97.8	95.2	**100.0**	84.4	90.3
DriveVLA-W0	1×C	98.7	**99.1**	95.3	99.3	83.3	90.2
Curious-VLA	1×C	98.4	96.9	97.9	98.1	**88.5**	90.3
Auto-JEPA (Ours)	1×C	98.4	98.3	95.0	**100.0**	87.1	**91.3**

$\mathrm{PDMS} = \mathrm{NC}\,\mathrm{DAC}\,\frac{5(\mathrm{EP}+\mathrm{TTC})+2\mathrm{C}}{12}$. We train on NAVSIM trainval and report results on the complete navtest split of 12,146 scenarios.

NAVSIM v2. NAVSIM v2 extends the evaluation to a broader set of driving-quality and rule-compliance properties. In addition to NC and DAC, it measures driving-direction compliance (DDC), traffic-light compliance (TL), ego progress (EP), time to collision (TTC), lane keeping (LK), history comfort (HC), and extended comfort (EC), and aggregates them into EPDMS. We evaluate v2 with the updated official implementation and human-behavior filtering enabled. Since v1 and v2 use different rollout and aggregation protocols, we report them separately without converting scores between the two metrics.

4.2 Implementation Details

The model takes four 256×256 front-camera frames, four historical ego positions, and a route command. Its 24-layer Transformer predictor has 16 heads and a hidden dimension of 1024, and outputs eight temporal latent tokens. We train with a per-GPU batch size of 8, learning rate 10^{-5}, weight decay 0.05, BF16 arithmetic, and an InfoNCE temperature of 0.07. The visual and target trajectory encoders remain frozen during predictor training. No object boxes, occupancy, semantic-map, or surrounding-agent motion labels are used. The scorer and gate use 75,823 training scenes and 8,459 validation scenes under a scene-prefix split; the validation and navtest sets have zero overlap in metric tokens. At inference, flat-cosine retrieval selects 300 candidates from a memory of 110,335 ground-truth trajectory–latent pairs; the scene scorer and feasibility gate then select the final trajectory using a DAC-failure threshold of 0.2.

Results on NAVSIM v1. As shown in Table 1, Auto-JEPA achieves **91.3 PDMS** using only a front camera, with 98.4 NC, 98.3 DAC, and 100.0 comfort.

Results on NAVSIM v2. Table 2 reports 85.6 EPDMS under the original evaluator and **89.1 EPDMS** under the updated official implementation with human-behavior filtering. CLOVER re-

Table 2: NAVSIM v2 comparison using each baseline's strongest reported backbone. For Auto-JEPA, † marks the updated official evaluation; unmarked rows use the original evaluation. Baseline references follow row order: (Chitta et al., 2023; Jiang et al., 2024; Liao et al., 2025; Li et al., 2025a; Yao et al., 2025; Li et al., 2026a;b; Chen et al., 2026; Jia et al., 2026).

Method	NC ↑	DAC ↑	DDC ↑	TL ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
End-to-End Planning Methods										
TransFuser	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
VADv2	97.3	91.7	98.2	**99.9**	77.6	92.7	66.0	**100.0**	**97.4**	76.6
DiffusionDrive	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
HydraMDP++ (ViT-L)	98.5	98.5	99.5	99.7	87.4	97.9	95.8	98.2	75.7	85.6
DriveSuprim (ViT-L)	98.4	98.6	**99.6**	99.8	**90.5**	97.8	**97.0**	98.3	78.6	87.1
VLA-Based Methods										
DriveVLA-W0	98.5	**99.1**	98.0	99.7	86.4	**98.1**	93.2	97.9	58.9	86.1
ReCogDrive	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6
Curious-VLA	98.4	96.9	99.2	99.8	88.5	97.9	96.9	98.1	81.5	85.3
DriveWorld-VLA	**98.6**	**99.1**	**99.6**	99.8	87.4	97.9	**97.0**	97.8	78.6	86.8
Auto-JEPA (Ours)	98.5	98.7	98.2	97.2	**90.5**	97.9	84.0	97.8	75.4	85.6
Auto-JEPA (Ours)†	98.5	98.7	98.3	99.7	90.5	97.9	94.7	97.8	75.2	**89.1**

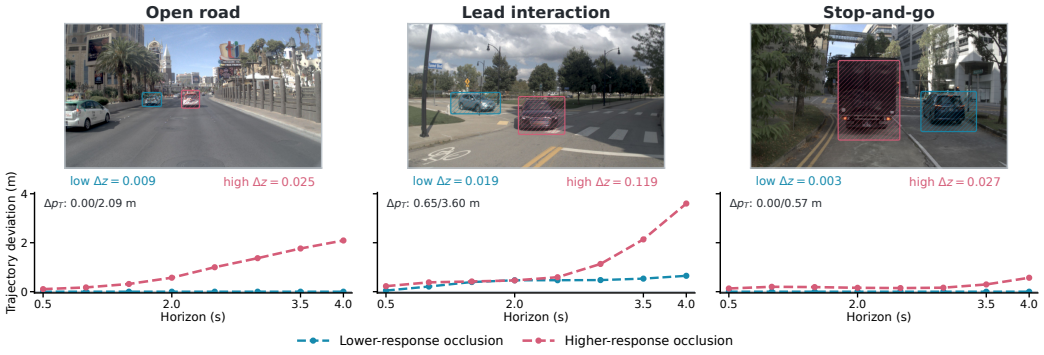


Figure 4: Selective responses to traffic participants. Cyan and rose denote occlusions of lower- and higher-impact vehicles, respectively. Curves show deviation from the unoccluded trajectory on a shared 0–4 m scale; Δp_T lists their terminal deviations in the same order. The smaller stop-and-go shift reflects its restricted motion range: the unoccluded plan moves only 0.17 m.

ports 90.4 EPDMS with a learned generator–scorer pipeline; Auto-JEPA remains competitive without parametric proposal generation. The evaluator change mainly affects TL and LK.

4.3 Ablation Studies

Component ablation. We ablate each component using the same Top-300 memory. Replacing the predicted intent with a fixed codebook medoid reduces PDMS from 91.3 to 52.6, confirming scene-conditioned retrieval. Intent-based cosine retrieval with the gate already reaches 87.6; adding the scorer contributes 3.7 points. Removing the gate yields 91.0 PDMS and lowers DAC from 98.3 to 97.9.

Candidate-pool size sensitivity. The system obtains 87.6, 91.1, and 91.3 PDMS for $K = 1$, 200, and 300. Increasing K from 1 to 200 improves PDMS by 3.5 points, demonstrating the value of candidate selection, whereas the marginal gain from 200 to 300 indicates that performance is approaching saturation.

Table 3: Component ablation on full NAVSIM v1 navtest. Checkmarks indicate enabled components.

Intent	Scorer	Gate	NC	DAC	TTC	C	EP	PDMS
✗	✓	✓	83.1	85.3	76.2	86.3	37.8	52.6
✓	✗	✓	98.1	96.4	94.0	**100.0**	81.7	87.6
✓	✓	✗	**98.5**	97.9	**95.0**	99.9	86.9	91.0
✓	✓	✓	98.4	**98.3**	**95.0**	**100.0**	**87.1**	**91.3**

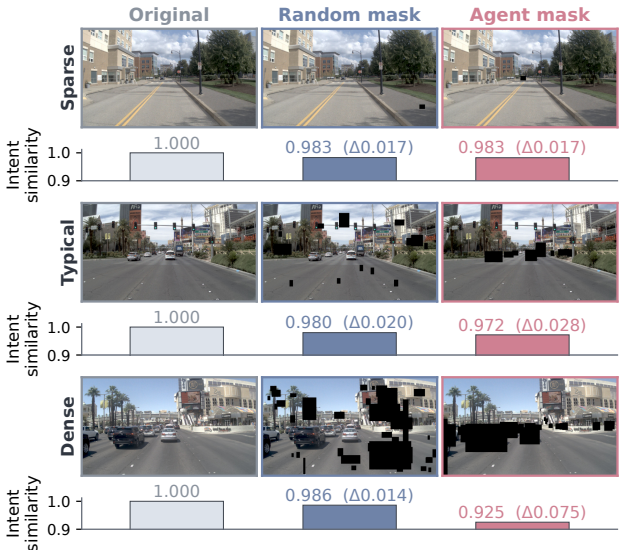


Figure 5: Representative controls from the full-validation semantic occlusion protocol. For each scene, dynamic-agent regions and independently sampled equal-area random regions are masked consistently across all four input frames. The bars report cosine similarity to the unoccluded intent.

4.4 Analysis

Selective sensitivity to dynamic-agent information. Across 15,364 validation samples, masking all dynamic agents in all four frames produces a mean intent change (1−cosine similarity) of 0.080, compared with 0.027 for independently sampled equal-area masks, a 2.97× increase; the dynamic-agent intervention is larger in 71.1% of samples. Figure 5 shows matched controls, while Figure 4 isolates individual vehicles. Vehicles affecting future driving cause larger latent and trajectory changes than low-impact vehicles. Differences grow with the horizon in the open-road and lead-interaction scenes, whereas the smaller stop-and-go response reflects an unoccluded plan that advances only 0.17 m. Although the predictor receives no object boxes, agent identities, interaction labels, or surrounding-agent motion annotations, these responses emerge from future ego-trajectory representation supervision. These results show that the model does not respond uniformly to all dynamic agents, but focuses more strongly on vehicles that may affect future driving decisions.

Conclusion. This paper introduced Auto-JEPA, an action-oriented latent world model that predicts continuous future driving intent. Joint-embedding prediction maps visual observations, ego-motion history, and navigation commands into the latent space of future ego trajectories; the predicted intent then retrieves executable candidates from a fixed memory for scene-conditioned selection. Auto-JEPA achieves 91.3 PDMS on NAVSIM v1 and 89.1 EPDMS on NAVSIM v2. Ablations and controlled occlusions validate the model's selective focus on scene features, supporting future-trajectory latent as a decision-relevant prediction target without dense future-world modeling. The current system remains limited by memory coverage and selection calibration; intent-conditioned trajectory generation or refinement provides a natural extension.

AI USE STATEMENT

OpenAI Codex was used for LaTeX template conversion, figure and table presentation, and formatting and reference checks.

REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. CLOVER: Closed-Loop Value Estimation and Ranking for End-to-End Autonomous Driving Planning. *arXiv preprint arXiv:2605.15120*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mahmoud 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.

Yuriy Biktairov, Maxim Stebelev, Irina Rudenko, Oleh Shliazhko, and Boris Yangel. PRANK: motion Prediction based on RANKing. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil Is in Narrow Policy: Unleashing Exploration in Driving VLA Models. *arXiv preprint arXiv:2603.06049*, 2026. Accepted to CVPR 2026 Findings.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. DrivingGPT: Unifying Driving World Modeling and Planning with Multi-Modal Autoregressive Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26890–26900, 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.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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*, volume 37, 2024.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. iPad: Iterative Proposal-Centric End-to-End Autonomous Driving. *arXiv preprint arXiv:2505.15111*, 2025.

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. *arXiv preprint arXiv:2309.17080*, 2023.

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, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning. *arXiv preprint arXiv:2402.13243*, 2024.

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*, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27137–27146, 2025c.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World Models Amplify Data Scaling Law in Autonomous Driving. In *International Conference on Learning Representations*, 2026a.

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*, 2026b.

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. *arXiv preprint arXiv:2406.06978*, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12037–12047, 2025.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi xin Yang. AdaThinkDrive: Adaptive Thinking via Reinforcement Learning for Autonomous Driving. *arXiv preprint arXiv:2509.13769*, 2025.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-Agent Motion Forecasting as Language Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8579–8590, 2023.

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. *arXiv preprint arXiv:2603.29163*, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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*, 2026.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drive-Dreamer: Towards Real-World-Driven World Models for Autonomous Driving. *arXiv preprint arXiv:2309.09777*, 2023a.

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. *arXiv preprint arXiv:2311.17918*, 2023b.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized Architecture for Real-Time Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15449–15458, 2024.

Chengen Xie, Bin Sun, Tianyu Li, Junjie Wu, Zhihui Hao, XianPeng Lang, and Hongyang Li. LatentVLA: Efficient Vision-Language Models for Autonomous Driving via Latent Action Prediction. *arXiv preprint arXiv:2601.05611v1*, 2026. URL https://arxiv.org/abs/2601.05611v1.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-Driven Flow Matching for Multimodal Trajectories Generation in End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1602–1611, 2025.

Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual Point Cloud Forecasting Enables Scalable Autonomous Driving. *arXiv preprint arXiv:2312.17655*, 2023.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards Precise Trajectory Selection for End-to-End Planning. *arXiv preprint arXiv:2506.06659*, 2025. Accepted to AAAI 2026.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive Diffusion World Model for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27220–27230, 2025.

Lingjun Zhang, Changjie Wu, Linzhe Shi, Jiangyang Li, Jiaxin Liu, Lei Yang, Hang Zhang, Mu Xu, and Hong Wang. DeepSight: Long-Horizon World Modeling via Latent States Prediction for End-to-End Autonomous Driving. *arXiv preprint arXiv:2605.10564*, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D Occupancy World Model for Autonomous Driving. *arXiv preprint arXiv:2311.16038*, 2023.

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*, pp. 28632–28642, 2025.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards End-to-end Autonomous Driving with Large Vision Language Action Model. *arXiv preprint arXiv:2503.23463*, 2025a.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025b.