ST-JEPA: JEPA-ENHANCED TEMPORAL SCENE–TRAJECTORY REPRESENTATION LEARNING FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end autonomous driving requires compact representations that preserve planning-relevant spatial and temporal context. A common design in recent ViT-based E2E planners is to introduce trajectory queries only after visual feature extraction, which limits the interaction between planning intent and visual representation learning during encoding. Meanwhile, in compact-token variants, scene tokens receive only auxiliary supervision without direct self-supervision on their latent representations. To address these limitations, we present **ST-JEPA**, a temporal scene–trajectory representation learning framework with two key designs: scene–trajectory encoding and JEPA-enhanced representation learning. First, ST-JEPA injects camera-aware scene tokens and ego-conditioned trajectory tokens directly into the visual encoder, enabling them to interact with image patches throughout feature extraction. Interleaved temporal global-register attention further aggregates cross-frame context and propagates the updated representations through subsequent backbone blocks. Second, ST-JEPA incorporates joint-embedding predictive learning through two complementary mechanisms. An online scene–trajectory JEPA objective predicts compact scene and trajectory representations from masked observations, using targets produced by an EMA teacher from unmasked inputs. This training-only objective provides latent self-supervision directly at the compact planning bottleneck. In parallel, we adapt JEPA pretraining to driving videos and integrate the resulting spatiotemporal latent features into the encoder by projecting them into the token space and fusing them with the native scene representation. ST-JEPA achieves state-of-the-art performance on NAVSIM-v1, NAVSIM-v2, and the closed-loop HUGSIM benchmark, with systematic ablations validating the effectiveness of the proposed scene–trajectory encoding, temporal interaction, and JEPA-based representation learning. The code will be made available upon publication.

1 INTRODUCTION

End-to-end (E2E) autonomous driving aims to directly predict future trajectories from sensor observations and ego states. A key challenge is to learn visual representations that preserve planning-relevant spatial and temporal context while remaining compact enough for efficient trajectory generation and scoring. A common design in recent ViT-based E2E planners is to introduce trajectory queries only after visual feature extraction (Kirby et al., 2026; Wang et al., 2026a), which limits the interaction between planning intent and visual representation learning during encoding. Compact-token planners such as DrivoR (Kirby et al., 2026) address the computational cost of multi-camera visual features by compressing them into a small set of camera-aware scene tokens for downstream planning. Despite its effectiveness, DrivoR uses a primarily single-frame visual representation and optimizes its compact scene tokens mainly through downstream trajectory and scoring objectives, without direct self-supervision on their latent representations.

To address these limitations, we present ST-JEPA, a JEPA-enhanced temporal scene–trajectory encoding framework for end-to-end planning. At the input of each frame-camera stream, ST-JEPA jointly injects camera-aware scene tokens and ego-conditioned trajectory tokens as prefix tokens. These tokens participate in the standard visual encoding process and interact with image patches

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.

inside the visual backbone. Rather than extracting frame-level tokens and fusing them after the backbone, ST-JEPA interleaves temporal fusion with visual encoding. At selected backbone depths, scene and trajectory tokens from the same camera are grouped across historical frames and updated through temporal global-register attention. The updated tokens are then written back to their original frame-camera positions and passed through the remaining backbone blocks. This in-backbone, per-camera temporal fusion allows the compact planning representation to combine camera identity, historical context, and ego-conditioned planning information.

The second design enhances the compact planning representation with JEPA through two complementary mechanisms. First, an online scene–trajectory JEPA objective uses masked visual observations to predict the current-frame features of both scene and trajectory tokens generated by an exponential-moving-average teacher from unmasked visual inputs. This training-only objective provides latent self-supervision directly at the compact planning bottleneck, without pixel-level reconstruction or additional inference-time computation. In parallel, we adapt JEPA pretraining to driving videos and integrate the resulting spatiotemporal latent features into the planner memory by projecting and aggregating them into compact scene tokens and fusing them with the task-adapted scene representation. This pretrained feature branch remains active at inference and provides driving-domain visual priors complementary to the task-adapted representation. Finally, we introduce a low-resolution visual prompt for both JEPA paths, supplying complementary coarse spatial cues during latent representation learning.

Overall, ST-JEPA unifies temporal scene–trajectory encoding with JEPA-based representation learning within a compact planning interface. By combining in-backbone temporal interaction, token-level latent supervision, and driving-video feature transfer, it learns a planning-oriented representation that is both temporally informed and directly aligned with downstream trajectory generation. Our contributions are summarized as follows:

1. We present ST-JEPA, a JEPA-enhanced temporal scene–trajectory representation learning framework that aligns compact visual encoding with end-to-end planning.

2. We introduce a temporal scene–trajectory encoder that injects camera-aware scene tokens and ego-conditioned trajectory tokens into the visual backbone, and performs per-camera temporal fusion inside the backbone through global-register attention.

3. We develop two complementary JEPA mechanisms: an online scene–trajectory JEPA objective that provides latent supervision directly to the compact planning tokens, and a driving-video V-JEPA 2 branch that projects and aggregates pretrained spatiotemporal features into additional scene tokens for planner memory.

4. We establish state-of-the-art performance in the reported benchmark comparisons: ST-JEPA reaches 94.85/95.37 PDMS on NAVSIM-v1 under the 85k and 103k training regimes, 91.8 EPDMS on NAVSIM-v2 Test, and 51.9 RC-Avg with 43.9 HD-Avg zero-shot on HUGSIM. Controlled ablations further validate the effectiveness of the proposed temporal encoding and complementary JEPA mechanisms.

2 Related work

2.1 Compact Scene Representations and Planning Queries for End-to-End Driving

End-to-end (E2E) autonomous driving maps sensor observations, ego state, and route context to a future ego trajectory. A central design question is how to expose rich visual observations to a planner without the cost of dense intermediate representations. Existing methods use different interfaces: UniAD (Hu et al., 2023) organizes perception and planning around shared queries, VAD (Jiang et al., 2023) and SparseDrive (Sun et al., 2025) use vectorized or sparse scene elements, and proposal-centric methods such as iPad (Guo et al., 2026) and DiffusionDrive (Liao et al., 2025) organize computation around candidate trajectories. These designs improve the structure or flexibility of planning, but often retain task-specific intermediate representations or introduce trajectory queries after visual feature extraction.

Perception-free planners instead compress image features into a small set of scene tokens consumed by a trajectory generator and/or scorer. DrivoR (Kirby et al., 2026) makes this interface explicit with camera-aware register tokens, which summarize each camera before lightweight trajectory-generation and scoring decoders. This substantially reduces downstream computation while preserving planning-relevant context. However, its registers are formed before trajectory queries are introduced, its observations are processed primarily frame by frame, and its compact scene tokens are optimized mainly through trajectory and scoring objectives. These limitations motivate compact representations in which planning information and temporal context can participate directly in visual encoding.

2.2 Temporal Modeling and Driving World Models

Temporal modeling in driving systems can be characterized by how information from multiple frames is combined and what compact representation is maintained for planning. BEVFormer (Li et al., 2024) uses spatial cross-attention and temporal self-attention to propagate camera features into a temporally aligned BEV representation. StreamPETR (Wang et al., 2023) instead adopts object-centric temporal modeling, maintaining a memory of object queries and propagating sparse instance-level representations across frames. These approaches establish two representative interfaces for temporal fusion: dense scene-level features and object-centric memory.

Planning-oriented models extend temporal representations toward end-to-end decision making. UniAD (Hu et al., 2023) shares a temporally enriched BEV representation across perception, prediction, and planning tasks. DriveWorld (Min et al., 2024) learns a memory-based 4D representation from multi-camera videos during pretraining, while World4Drive (Zheng et al., 2025) combines observations in a physical latent encoder to support intention-aware trajectory selection. Together, these methods show that temporal information can improve planning when it is retained in a structured scene, object, or latent representation rather than treated as independent single-frame features.

ST-JEPA places this interaction inside the visual backbone: same-camera scene and trajectory tokens are updated at selected depths before continuing through later blocks. This differs from memory, BEV, and object-query-based interfaces while retaining the compact scene-token interface of the planner.

2.3 JEPA-Based Predictive Representation Learning for Driving

Joint-Embedding Predictive Architectures (JEPAs) predict target embeddings rather than pixels. I-JEPA (Assran et al., 2023) establishes this principle for images, V-JEPA (Bardes et al., 2024) extends it to video, and V-JEPA 2 (Assran et al., 2025) demonstrates transfer to visual understanding, future prediction, and planning. This family provides semantic and temporal supervision in feature space while avoiding generative reconstruction.

Driving methods use JEPA with different targets and roles. Drive-JEPA (Wang et al., 2026a) adapts V-JEPA to driving videos and transfers its representation to an E2E planner with multimodal trajectory distillation.

ST-JEPA extends driving-video JEPA with an online objective at the compact planning bottleneck and an inference-time driving-video feature branch. The former provides training-only token-level latent supervision, while the latter supplies complementary driving-domain features.

3 Method

3.1 Framework Overview

ST-JEPA is a compact-token end-to-end planner built on the DrivoR (Kirby et al., 2026) planning interface. Given a temporal sequence of multi-camera observations, the ego state, and route-conditioned inputs, it constructs a temporal scene–trajectory representation and uses the original DrivoR trajectory decoder and scorer to generate and rank candidate ego trajectories. Our changes are confined to representation construction and JEPA supervision; the downstream planning interface and objectives remain unchanged.

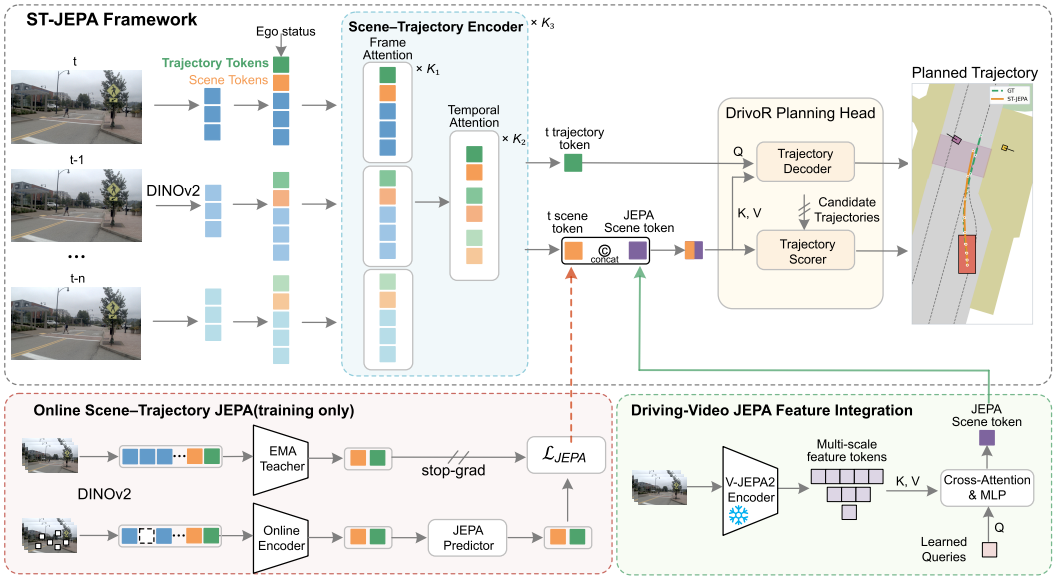


Figure 1: **Overview of ST-JEPA.** ST-JEPA performs in-backbone temporal scene–trajectory encoding to construct a compact planning representation. A training-only scene–trajectory JEPA objective provides latent supervision, while a driving-domain-adapted V-JEPA 2 branch introduces complementary spatiotemporal features. The resulting representation is consumed by the original DrivoR planner.

Let $I_{t-T+1:t}$ denote T observed frames from N cameras. The temporal encoder processes the complete temporal sequence, while the planner consumes only the tokens associated with the latest frame. ST-JEPA consists of (i) a temporal scene–trajectory encoder, (ii) a training-only scene–trajectory JEPA objective, and (iii) an driving-domain-adapted V-JEPA 2 (Assran et al., 2025) scene-token branch.

3.2 Temporal Scene–Trajectory Encoder

For camera n, the encoder maintains K_s learnable scene tokens, $S_n^{(0)} \in \mathbb{R}^{K_s \times D_b}$, where D_b is the visual-backbone width. It also constructs K_q trajectory tokens from a learnable seed, the current ego state s_t, and a camera embedding:

$$Q_{\tau,n}^{(0)} = Q_0 + \text{reshape}(\phi(s_t)) + E_n. \tag{1}$$

Here, ϕ projects the ego state to $K_q D_b$ dimensions and E_n identifies the camera. The same planning-conditioned token structure is instantiated for every observed frame. Scene and trajectory tokens are concatenated with image patches before the first visual block, so both token types participate in visual feature formation rather than being introduced only after image encoding.

To fuse time inside the backbone, ST-JEPA inserts temporal global-register attention after selected visual blocks. For camera n at depth l, tokens from all frames are collected as

$$X_n^{(l)} \in \mathbb{R}^{T(K_s+K_q)\times D_b}. \tag{2}$$

A learnable temporal position embedding and self-attention update $X_n^{(l)}$ independently for each camera. The updated tokens are written back to their original frame-camera locations and continue through the remaining visual blocks. This preserves camera identity while allowing corresponding scene and trajectory tokens to exchange information across time.

After the backbone neck, tokens are projected to planner dimension D and form $Z_{1:T}^s = \{Z_{\tau,n}^s\}$ and $Z_{1:T}^q = \{Z_{\tau,n}^q\}$. The planner selects the latest-frame representations:

$$Z_t^s = \text{select}_{\tau=T}(Z_{1:T}^s), \qquad Z_t^q = \text{select}_{\tau=T}(Z_{1:T}^q). \tag{3}$$

The scene tokens Z_t^s provide visual memory for the action-conditioned planner. A lightweight split/projection layer expands Z_t^q into the fixed number of proposal tokens required by the original DrivoR (Kirby et al., 2026) trajectory decoder. Consequently, the decoder and scorer keep their original interfaces and planning losses, while the temporal observations have already influenced the current compact planning representation.

3.3 JEPA-Based Representation Enhancement

ST-JEPA uses JEPA in two complementary ways. The first regularizes the task-adapted compact scene–trajectory representation during training. The second transfers complementary spatiotemporal features learned from driving videos.

3.3.1 Training-Only Scene–Trajectory JEPA

Trajectory supervision is sparse relative to the visual information in the input sequence. We therefore apply a JEPA objective directly to the compact scene–trajectory representation consumed by the planner. The online encoder E_θ receives a masked sequence, while an EMA teacher $E_{\bar{\theta}}$ receives the unmasked sequence. Because both token types are formed by interacting with the patch sequence inside the backbone, masked positions are retained to preserve the pretrained token layout. For a patch position i,

$$\widetilde{p}_i = \begin{cases} p_i, & i \notin \mathcal{M}, \\ e_{\text{mask}}, & i \in \mathcal{M}, \end{cases} \qquad \widetilde{q}_j = \begin{cases} q_j + \bar{q}_j^{\text{lr}}, & j \in \mathcal{T}_{\text{lr}}, \\ q_j, & j \notin \mathcal{T}_{\text{lr}}. \end{cases} \tag{4}$$

Here, $\mathcal{M}$ is the selected patch-mask set and $\mathcal{T}_{\text{lr}}$ is a sampled subset of target-token positions. The low-resolution prompt is obtained by downsampling the image, applying the corresponding embedding, and projecting it to the target-token dimension. ST-JEPA uses one target-token prompt design across its two JEPA paths. During pretraining, the planning representation path forms target tokens inside the encoder and injects the prompt during encoder-side processing; the driving-video representation path forms target tokens at the predictor interface and injects the prompt there. In both pretraining paths, the prompt is added to only a subset of target tokens, with reported coverage approximately $|\mathcal{T}_{\text{lr}}|/|\mathcal{T}| \approx 0.7$, rather than to all positions.

For the latest frame, the online and teacher representations retain both token types:

$$U_t = \text{select}_t\Big(E_\theta(\widetilde{I}_{t-T+1:t}, s_t)\Big), \tag{5}$$

$$V_t = \text{sg}[\text{select}_t(E_{\bar{\theta}}(I_{t-T+1:t}, s_t))]. \tag{6}$$

The teacher is updated by exponential moving average,

$$\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta. \tag{7}$$

A lightweight predictor P_ψ maps the online scene–trajectory tokens to the teacher-token space using learnable target tokens and self-attention. The teacher tokens are targets only and are not supplied to the predictor:

$$\widehat{V}_t = P_\psi(U_t), \qquad \mathcal{L}_{\text{JEPA}} = \frac{1}{MD}\left\|\widehat{V}_t - \text{LN}(V_t)\right\|_1, \tag{8}$$

where $M = N(K_s + K_q)$ and D is the planner feature dimension. Unlike pixel reconstruction or dense patch prediction, this loss directly regularizes the compact scene–trajectory representation used for planning. The predictor and EMA teacher are removed at inference.

3.3.2 Driving-Video Feature Integration

The second path adapts a V-JEPA 2 (Assran et al., 2025) encoder on driving videos and converts its multi-scale spatiotemporal features into a compact scene-token memory. Let F_l^{VJ} denote the feature sequence from scale or backbone depth l. Each scale is projected to the planner dimension and receives a scale embedding:

$$\widetilde{F}_l = W_l F_l^{\text{VJ}} + e_l, \qquad \widetilde{F}^{\text{VJ}} = \text{concat}_{l=1}^{L}(\widetilde{F}_l). \tag{9}$$

A learnable query bank $Q_0 \in \mathbb{R}^{K_{\text{VJ}} \times D}$ attends to the projected multi-scale memory through a V-JEPA 2 query aggregation module to produce a fixed number of V-JEPA 2 scene tokens:

$$Z_{\text{scene}}^{\text{VJ}} = \text{QueryAgg}(Q_0, \widetilde{F}^{\text{VJ}}). \tag{10}$$

The low-resolution prompt is used only during V-JEPA 2 pretraining and is not part of the inference-time query aggregation shown here. The resulting V-JEPA 2 scene tokens and the task-adapted scene memory are given source embeddings and fused as

$$F_t^{\mathrm{ST}} = Z_t^s + e_{\mathrm{ST}}, \qquad F^{\mathrm{VJ}} = Z_{\mathrm{scene}}^{\mathrm{VJ}} + e_{\mathrm{VJ}}, \qquad Z_t^{\mathrm{mem}} = \mathrm{concat}(F_t^{\mathrm{ST}}, F^{\mathrm{VJ}}). \tag{11}$$

The adapted V-JEPA 2 encoder is frozen; the multi-scale projections, query bank, query aggregation, and source embeddings are learned by the planner. This branch therefore contributes a compact, fixed-size scene representation rather than directly appending all dense video tokens.

3.4 Planning and Training Objective

During training, the online branch is evaluated on the masked temporal sequence and the EMA branch on the corresponding unmasked temporal sequence. The planning heads use the current-frame representation, while the JEPA loss supplies an additional latent-level constraint. The current-frame scene memory and trajectory-derived proposal tokens are passed to the unchanged DrivoR (Kirby et al., 2026) trajectory decoder and scorer. The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{plan}} + \lambda_{\mathrm{JEPA}} \mathcal{L}_{\mathrm{JEPA}}, \tag{12}$$

where $\mathcal{L}_{\mathrm{plan}}$ denotes the original proposal and trajectory-scoring losses. During inference, the input temporal sequence is encoded once, the latest-frame tokens are selected, and the decoder ranks the resulting candidate trajectories. The training-only JEPA predictor and EMA teacher are absent; when enabled, the frozen driving-video encoder remains in the inference path.

4 Experiments

4.1 Experimental Setup

We evaluate on NAVSIM-v1 (Dauner et al., 2024) and the standard NAVSIM-v2 (Cao et al., 2025) Test protocol using the public `navtest` split; zero-shot transfer is additionally measured on the closed-loop HUGSIM benchmark (Zhou et al., 2025). NAVSIM-v1 uses 85k-sample `navtrain`, 103k-sample `trainval` training settings, while ablations use `navtrain`. We report the benchmark sub-metrics and the Predictive Driver Model Score (PDMS) (Dauner et al., 2024),

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5(\mathrm{EP} + \mathrm{TTC}) + 2\mathrm{C}}{12}. \tag{13}$$

We compare with published camera-only planners and DrivoR (Kirby et al., 2026). Published values are reference comparisons because sensors, splits, evaluator versions, and auxiliary supervision can differ; reproduced DrivoR and ST-JEPA ablations share the same code and evaluation pipeline. Detailed architecture, optimization, and video-encoder settings are provided in Appendix A.1. Checkpoints are selected on validation PDMS and evaluated once on `navtest`; values marked with an asterisk are matched ablation runs.

4.2 Main Results on NAVSIM-v1

We evaluate NAVSIM-v1 (Dauner et al., 2024) on the public `navtest` split using the PDMS metric. The main comparison reports the benchmark sub-metrics: no-at-fault collision (NC), drivable-area compliance (DAC), time-to-collision compliance (TTC), comfort (C), and ego progress (EP), with results summarized in Table 1. The training-data column identifies the reported NAVSIM-v1 training scale; "–" denotes a non-learned baseline or human reference without a training split. For rows where only PDMS was reported, the sub-metrics are shown as "–" rather than inferred. The DrivoR (Kirby et al., 2026) rows are reported results under the listed training-data settings, including the 85k result of 93.13 PDMS. Since published methods may use different implementations, evaluator versions, or auxiliary supervision, the table should be interpreted as a reference comparison rather than a strictly controlled reproduction.

Under the corresponding training-data settings, ST-JEPA improves over the reported DrivoR (Kirby et al., 2026) results from 93.13 to 94.85 PDMS at 85k samples, from 93.7 to 95.37 at 103k samples. These gains of 1.72, 1.67 PDMS, respectively, indicate that the proposed scene–trajectory encoding and JEPA-based representation learning provide benefits beyond simply increasing the training data.

Table 1: NAVSIM-v1 results on `navtest`. Training-data labels are reported by the corresponding methods; higher is better. Best results among learned methods are bolded; tied best results are all highlighted, while the human-driver reference is excluded from ranking.

Method	Training data	NC	DAC	TTC	C	EP	**PDMS**
PDM-Closed (Dauner et al., 2023)	–	94.60	99.80	89.90	86.90	99.90	89.10
Human driver (Dauner et al., 2024)	–	100.00	100.00	100.00	99.90	87.50	94.80
RAP-DINO (Feng et al., 2026)	10× navtrain†	99.10	98.90	96.70	**100.00**	90.30	93.80
Ego-stat. MLP (Dauner et al., 2024)	Trainval 103k	93.00	77.30	83.60	**100.00**	62.80	65.60
UniVLA (Wang et al., 2026b)	Trainval 103k	96.90	91.10	91.70	96.70	76.80	81.70
DrivingGPT (Chen et al., 2025)	Trainval 103k	98.90	90.70	94.90	95.60	79.70	82.40
UniAD (Hu et al., 2023)	Trainval 103k	97.80	91.90	92.90	**100.00**	78.80	83.40
LTF (Chitta et al., 2022)	Trainval 103k	97.40	92.80	92.40	**100.00**	79.00	83.80
PARA-Drive (Weng et al., 2024)	Trainval 103k	97.90	92.40	93.00	99.80	79.30	84.00
DriveX-S (Shi et al., 2025)	Trainval 103k	97.50	94.00	93.00	**100.00**	79.70	84.50
World4Drive (Zheng et al., 2025)	Trainval 103k	97.40	94.30	92.80	**100.00**	79.90	85.10
DRAMA (Yuan et al., 2024)	Trainval 103k	98.00	93.10	94.80	**100.00**	80.10	85.50
VADv2 (Chen et al., 2024)	Trainval 103k	98.10	94.80	94.30	**100.00**	80.60	86.20
PRIX (Wozniak et al., 2026)	Trainval 103k	98.10	96.30	94.10	**100.00**	82.30	87.80
DiffusionDrive (Liao et al., 2025)	Trainval 103k	98.20	96.20	94.70	**100.00**	82.20	88.10
DIVER (Song et al., 2025)	Trainval 103k	98.50	96.50	94.90	**100.00**	82.60	88.30
DriveVLA-W0 (Li et al., 2026)	Trainval 103k	98.70	99.10	95.30	99.30	83.30	90.20
ReCogDrive (Xiong et al., 2026)	Trainval 103k	97.90	97.30	94.90	**100.00**	87.30	90.80
Hydra-MDP++ (Li et al., 2025)	Trainval 103k	98.60	98.60	95.10	**100.00**	85.70	91.00
iPad (Guo et al., 2026)	Trainval 103k	98.60	98.30	94.90	**100.00**	88.00	91.70
DriveSuprim (Yao et al., 2026)	Trainval 103k	98.60	98.60	95.50	**100.00**	91.30	93.50
Drive-JEPA (Wang et al., 2026a)	Trainval 103k	–	–	–	–	–	93.70
DrivoR (Kirby et al., 2026)	Train 85k	98.93	98.35	96.25	**100.00**	89.92	93.13
	Trainval 103k	99.00	98.90	96.70	**100.00**	90.00	93.70
ST-JEPA (Ours)	Train 85k	99.07	98.84	96.41	**100.00**	93.22	94.85
	Trainval 103k	**99.22**	**99.19**	**96.74**	**100.00**	**93.59**	**95.37**

Table 2: Standard NAVSIM-v2 test results. Higher is better.

Method	NC	DAC	DDC	TL	EP	TTC	LK	HC	EC	**EPDMS**
LTF (Chitta et al., 2022)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
VADv2 (Chen et al., 2024)	97.3	91.7	98.2	**99.9**	77.6	92.7	66.0	**100.0**	**97.4**	76.6
DiffusionDrive (Liao et al., 2025)	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
Hydra-MDP++ (Li et al., 2025)	98.5	98.5	99.5	99.7	87.4	97.9	95.8	98.2	75.7	85.6
DriveSuprim (Yao et al., 2026)	98.4	98.6	99.6	99.8	90.5	97.8	97.0	98.3	78.6	87.1
DriveVLA-W0 (Li et al., 2026)	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
ReCogDrive (Xiong et al., 2026)	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6
DriveWorld-VLA (Liu et al., 2026)	98.6	99.1	99.6	99.8	87.4	97.9	97.0	97.8	78.6	86.8
Drive-JEPA (Wang et al., 2026a)	98.4	98.6	99.1	99.8	88.4	97.8	**97.6**	97.9	84.8	87.8
DrivoR (Kirby et al., 2026)	99.2	99.0	99.7	99.6	89.6	98.7	95.6	98.0	73.9	90.5
ST-JEPA (Ours)	**99.4**	**99.4**	**99.8**	99.4	**92.4**	**99.0**	96.7	97.5	70.3	**91.8**

4.3 Main Results on NAVSIM-v2 Test

Following Drive-JEPA (Wang et al., 2026a), we evaluate ST-JEPA under the standard NAVSIM-v2 test protocol and report NC, DAC, DDC, TL, EP, TTC, LK, HC, EC, and EPDMS. The results are presented in Table 2. Best result in each column is shown in bold, with all tied best results highlighted. Results for other methods are source-reported. ST-JEPA achieves the highest EPDMS among the listed methods at 91.8, improving over DrivoR (90.5) by 1.3 EPDMS.

4.4 Ablation Studies

We use the matched 85k setting for all ablations and keep the planner, optimizer, and evaluation pipeline fixed. Table 3 combines the cumulative component study with targeted temporal and JEPA controls. The cumulative path improves from 92.75 to 94.85 PDMS: temporal global-register attention, trajectory tokens, online token-level JEPA, and driving-video JEPA contribute complementary gains. The targeted controls further show that in-backbone fusion is preferable to post-backbone fusion, compact-token prediction is preferable to patch-level prediction, and driving-specific pre-training is more useful than generic V-JEPA. The online teacher and predictor are removed after

Table 3: Ablation studies on NAVSIM-v1 under the matched 85k training setting. Δ is measured relative to the control row of each study block. Rows marked with * denote alternative comparison modules that are evaluated only for ablation and are not used in the final ST-JEPA model.

Study	Variant	PDMS	Δ vs. control
Cumulative	Reproduced DrivoR (Kirby et al., 2026) baseline	92.75	–
	+ temporal global-register attention	93.23	+0.48
	+ in-backbone trajectory tokens	93.55	+0.80
	+ online token-level JEPA	94.08	+1.33
	+ driving-video JEPA and prompt	**94.85**	+2.10
Temporal fusion	Reproduced DrivoR (Kirby et al., 2026) baseline	92.75	–
	In-backbone global-register attention	**93.23**	+0.48
Online JEPA	+ in-backbone trajectory tokens	93.55	–
	Patch-level target*	93.70	+0.15
	Scene/trajectory-token target	93.88	+0.33
	Token target + low-resolution prompt	**94.08**	+0.53
Video JEPA	Token target + low-resolution prompt	94.08	-
	Standard ViT-L*	94.13	+0.05
	Generic V-JEPA ViT-L*	94.15	+0.07
	Driving-video V-JEPA ViT-L	94.65	+0.57
	Driving-video V-JEPA + low-resolution prompt	**94.85**	+0.77

training, whereas the frozen video branch remains active at inference. Detailed per-study protocols and full ablation tables are provided in Appendix A.2.

The detailed ablations support three practical conclusions: temporal fusion should occur inside the visual backbone, JEPA targets should be defined on the compact scene–trajectory tokens rather than patches, and driving-domain JEPA pretraining is more valuable than generic video initialization.

4.5 Qualitative Comparison on NAVSIM

As shown in Fig. 2, the examples illustrate how ST-JEPA improves planning in situations that require temporal scene understanding and explicit reasoning about road structure and driving constraints. In the first row, ST-JEPA adopts a more conservative braking behavior in response to a potential risk, whereas DrivoR continues with a substantially larger forward displacement. In the second row, the DrivoR trajectory exhibits a pronounced lateral deviation and fails to maintain the correct drivable area and road direction. ST-JEPA remains aligned with the valid driving corridor and follows the intended direction.

The third row focuses on collision avoidance and continued progress. Although DrivoR produces a forward-extending trajectory, it does not simultaneously satisfy the collision-safety and effective-progress requirements. ST-JEPA avoids the risk while preserving continuous forward motion. The fourth row shows a turning scenario. DrivoR remains within the drivable area but makes insufficient progress after the turn, whereas ST-JEPA completes the turn and continues along the subsequent road segment.

Overall, the qualitative results in Fig. 2 suggest that ST-JEPA produces planning behaviors that better account for potential risks, drivable-area constraints, road direction, and progress through turns. Quantitative comparisons are reported separately using the NAVSIM metrics.

4.6 HUGSIM Generalization

We evaluate ST-JEPA zero-shot on the 345-scenario HUGSIM (Zhou et al., 2025) test set without HUGSIM-specific fine-tuning. Table 4 reports average Road Completion (RC) and HUGSIM Driving Score (HD-Score). ST-JEPA improves over DrivoR (Kirby et al., 2026) by 2.1 RC points and 8.2 HD-Score points.

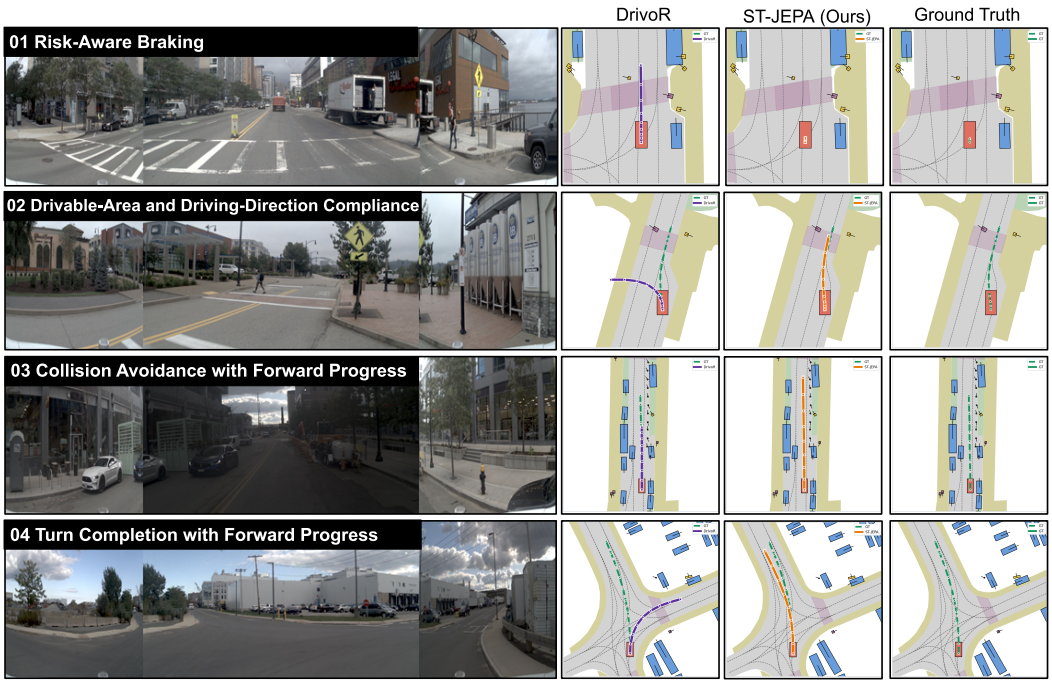


Figure 2: **Qualitative planning comparison on representative NAVSIM-v1 test scenes.** Each row shows one driving scenario. The left panel presents the current multi-camera RGB observation, followed by bird's-eye-view planning results from DrivoR, ST-JEPA, and the ground-truth future trajectory. The planning horizon is 4 s with eight poses sampled at 0.5 s intervals. Colored solid curves denote model predictions, while the dashed curve denotes the ground-truth trajectory.

Table 4: Zero-shot HUGSIM generalization. Higher is better.

Method	RC-Avg	HD-Avg
VAD (Jiang et al., 2023)	31.4	13.4
LTF (Chitta et al., 2022)	38.9	23.7
UniAD (Hu et al., 2023)	45.9	32.7
DrivoR (Kirby et al., 2026)	49.8	35.7
ST-JEPA	**51.9**	**43.9**

5 Conclusion

We present ST-JEPA, a temporal scene–trajectory representation learning framework with two key designs: scene–trajectory encoding and JEPA-enhanced representation learning. The online token-level objective regularizes the planner bottleneck during training, while the driving-video JEPA features provide complementary spatiotemporal information; both are integrated without changing the trajectory-decoder interface.

Across NAVSIM-v1 training scales, ST-JEPA reaches 95.37 PDMS, obtains 91.8 EPDMS on NAVSIM-v2 Test, and improves zero-shot transfer on HUGSIM over DrivoR. Under the corresponding benchmark protocols, these results establish state-of-the-art performance across the reported NAVSIM-v1, NAVSIM-v2, and HUGSIM comparisons. They support aligning temporal visual encoding and latent representation learning with planning tokens to improve compact representations for end-to-end driving.

AI USE STATEMENT

We used LLMs to help draft, edit, and polish portions of the manuscript, including improving clarity, organization, and presentation. We have reviewed and verified all technical content, experimental results, analyses, and claims. We take full responsibility for the paper's content.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training objectives, implementation settings, and evaluation protocols in the main paper and appendix. Upon acceptance, we will publicly release the complete source code, including the training and inference configurations, to facilitate 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, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. 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.

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.

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

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26890–26900. IEEE, 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, 2022.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pp. 1268–1281. PMLR, 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.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 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*, 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.

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. IEEE, 2023.

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.

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*, 2025.

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, 2026.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2024.

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.

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

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.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation. *arXiv e-prints*, pp. arXiv–2507, 2025.

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.

Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8372–8381. IEEE, 2023.

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*, 2026a.

Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3598–3608. IEEE, 2023.

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, 2026b.

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

Maciej Wozniak, Lianhang Liu, Yixi Cai, and Patric Jensfelt. Prix: Learning to plan from raw pixels for end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(5):6400–6407, 2026.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026.

Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 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*, 2025.

Table 5: Input and output configurations of the two backbones.

Configuration	DINOv2 ViT-S/14	Driving-adapted V-JEPA2 ViT-L
Camera input	Front, front-left, front-right, rear	Front only
Temporal input	4 frames at 2 Hz	Same 4-frame sequence
Resolution	1148×672	256×512
Output to planner	64 scene and 64 trajectory tokens	128 scene tokens from query aggregation

Table 6: Data splits and evaluation protocols used in the experiments.

Dataset or setting	Role	Scale or protocol
NAVSIM-v1 `navtrain`	Planner training and ablations	85k samples
NAVSIM-v1 `trainval`	Data-scaling comparison	103k samples
NAVSIM `navtest`	Held-out evaluation	12,146 valid scenarios
NAVSIM-v2 Test	Final EPDMS evaluation	Standard test protocol
HUGSIM	Zero-shot transfer	345 scenarios

A Appendix

A.1 Additional Experimental Details

This supplement provides the input and output specifications of the two backbones and a focused discussion of the ablation results. Dataset definitions, evaluation metrics, and main benchmark comparisons are reported in the main paper and are not repeated here.

(1) Inputs and Outputs of the Two Backbones. ST-JEPA combines a task-adapted DINOv2 ViT-S backbone (Oquab et al., 2023) with a frozen, driving-adapted V-JEPA2 ViT-L backbone. The two backbones observe the same temporal interval but use different camera coverage and spatial resolution.

The DINOv2 ViT-S backbone is the primary planning representation. Its latest-frame output contains 64 scene tokens and 64 trajectory tokens for the DrivoR decoder. The V-JEPA2 backbone processes only the front-camera sequence; its multi-scale features are projected to the planner dimension and compressed by the learned query aggregation module into 128 scene tokens. These tokens are fused with the DINOv2 scene memory, while the V-JEPA2 backbone remains frozen during planner training.

The V-JEPA2 backbone is adapted before planner training using 8-frame driving clips sampled at 2 FPS, a temporal tubelet size of 2, and spatiotemporal block masking. The low-resolution prompt is used only during JEPA pretraining. It is not part of the four-frame inference input and is not injected into the inference-time query aggregation module.

(2). Data Splits and Evaluation Protocol. The main paper defines the datasets and metrics; this subsection records the split conventions used for the controlled experiments.

Unless stated otherwise, all component ablations use the matched 85k `navtrain` setting. The driving-video JEPA pretraining clips are kept separate from the OpenScene (Tong et al., 2023) trainval supervised planner data; the resulting video encoder is frozen during planner optimization.

A.2 Discussion of Ablation Results

Temporal interaction. As shown in Table 7, the reproduced single-frame DrivoR baseline obtains 92.75 PDMS. Post-backbone concatenation and post-backbone learnable-query fusion reach 93.01 and 92.90, respectively, whereas in-backbone global-register attention reaches 93.23. The result indicates that temporal information is more effective when it participates in visual representation formation than when independently encoded frame features are fused only at the planner input.

Trajectory-conditioned representation. As shown in Table 3, adding ego-conditioned trajectory tokens inside the temporal visual backbone increases PDMS from 93.23 to 93.55. This gain supports introducing action-related conditions while the visual representation is being formed, instead of adding trajectory queries only after image encoding.

Table 7: Comparison of temporal interaction strategies under the matched 85k training setting.

Temporal interaction strategy	Input frames	Fusion location	PDMS
Single-frame scene tokens	1	No temporal fusion	92.75
Post-backbone concatenation	4	After independent visual encoding	93.01
Post-backbone learnable-query attention	4	After independent visual encoding	92.90
In-backbone global-register attention (ours)	4	At selected backbone depths	**93.23**

Online scene–trajectory JEPA. As shown in Table 3, starting from the 93.55 temporal scene–trajectory encoder, patch-level JEPA reaches 93.70, joint scene- and trajectory-token prediction reaches 93.88, and the same compact-token target with the low-resolution pretraining prompt reaches 94.08. Directly regularizing the compact tokens consumed by the planner is therefore more effective than predicting dense patch features. The prompt supplies an additional 0.20-point gain by providing coarse spatial evidence during training.

Driving-video JEPA. As shown in Table 3, with the online JEPA model at 94.08 as the reference, an image-pretrained ViT-L and a generic V-JEPA encoder reach 94.13 and 94.15, while driving-video JEPA pretraining raises the result to 94.65. This comparison first verifies the value of driving-domain JEPA pretraining rather than model capacity alone. We then study how the pretrained representation can be integrated into the current end-to-end planner: multi-scale V-JEPA2 features are projected and aggregated into 128 scene tokens before fusion with the native planner memory. Finally, introducing the low-resolution prompt during driving-video JEPA pretraining further improves PDMS from 94.65 to 94.85. These results show that both driving-domain pretraining and its planner-side integration are effective, while the prompt provides an additional gain. Overall, the complete model improves the matched 92.75 baseline by 2.10 PDMS.

A.3 Limitations

The driving-video branch uses only front-camera history, so its pretrained spatiotemporal representation does not directly cover side and rear views. Extending driving-video pretraining and feature integration to multi-camera temporal observations is left for future work.