DA-WAM: DECISION-ALIGNED FUTURE LATENTS FOR DRIVING WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models for autonomous driving should anticipate the consequences that matter for driving decisions rather than reconstruct every detail of future scenes. However, predictive representations alone are not guaranteed to distinguish candidate actions that lead to different planning outcomes. We propose DA-WAM, a decision-aligned world action model that couples a Joint-Embedding Predictive Architecture (JEPA) with candidate-level trajectory scoring. During joint training, the JEPA objective grounds future representations in observed scene dynamics, while the planning objectives encourage these representations to capture distinctions relevant to safety and utility. An action-conditioned predictor generates a distinct future latent for each candidate trajectory, and a multi-head trajectory scorer uses the corresponding latent to estimate planning factors and overall utility. In addition, safety-critical post-training with hard negatives further improves discrimination between geometrically similar trajectories with different safety outcomes. DA-WAM achieves state-of-the-art performance among learned camera-based planners on NAVSIM-v1, NAVSIM-v2, and Bench2Drive. Ablation studies demonstrate the effectiveness of each component in driving decision-making.

1 INTRODUCTION

Safe and effective autonomous driving requires reasoning about how the scene would evolve under each candidate ego action. World models address this challenge by predicting future visual or latent states from current observations and proposed actions Wang et al. (2024); Li et al. (2025b); Zheng et al. (2025). Yet the fundamental question is not simply whether a model can predict the future, but whether its predictions are decision-informative: do the predicted consequences of each candidate action directly determine how that trajectory is evaluated? To realize the full potential of world models, future prediction must therefore be tightly coupled with decision making, such that each candidate is evaluated against the future predicted specifically for that action.

Existing approaches pursue future prediction along two broad directions, but neither direction alone guarantees this coupling. Methods such as DriveWorld Min et al. (2024), LAW Li et al. (2025b), Drive-JEPA Wang et al. (2026c), and Latent-WAM Wang et al. (2026b) use temporal prediction primarily to strengthen scene representations or trajectory learning. Although these approaches learn rich predictive features, predictive pretraining and planner optimization are often stage-separated; fixed or frozen encoders consequently cannot adapt their future representations to the specific demands of trajectory scoring. A second line of work brings predicted futures more directly into planning. Drive-WM Wang et al. (2024) generates future views for planning, WoTE Li et al. (2025c) evaluates trajectories through a BEV world model, World4Drive Zheng et al. (2025) reasons over intention-conditioned latent futures, and DriveFuture Hong et al. (2026) conditions its planner on a predicted future latent. These methods move closer to using future predictions at decision time, yet predicted states may still be shared, pooled, or only weakly associated with individual candidates. As summarized in Fig. 1, these designs weaken the correspondence between an action and its predicted consequence. The scorer may therefore rely primarily on geometric cues rather than the scene-conditioned future content that distinguishes safe from unsafe outcomes.

We argue that the planning value of a world model is bounded by how directly its predictions influence candidate-level scoring. Evaluating each candidate against its own predicted future would allow the planner to exploit action-specific consequences, such as collisions, lane departures, or

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

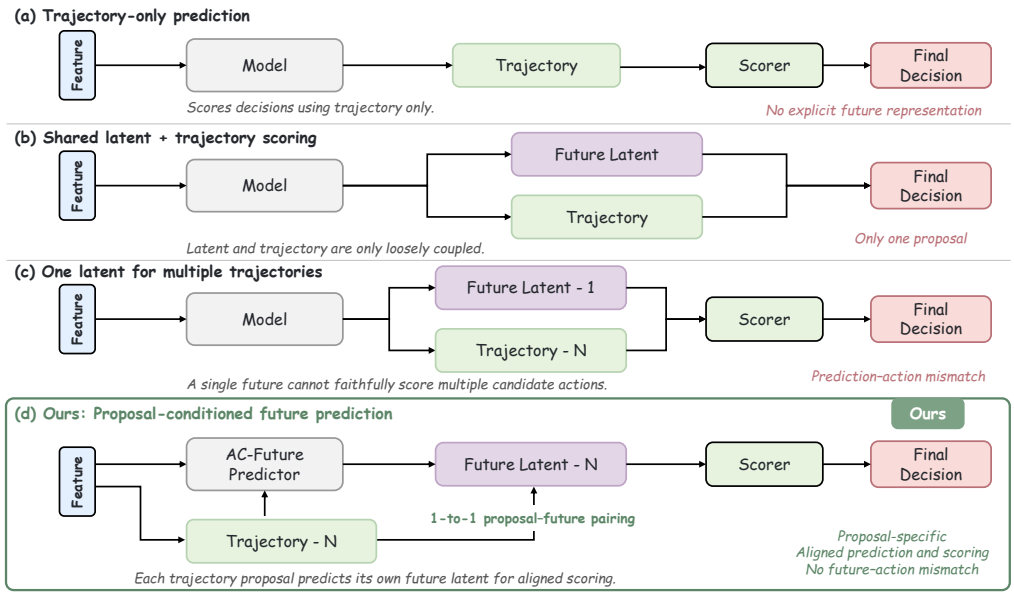

Figure 1: **Candidate–representation coupling in driving decisions.** Schematic comparison of (a) trajectory prediction without an explicit future-representation input to the scorer, (b) joint trajectory and latent prediction without explicit candidate-wise pairing, (c) multiple candidates evaluated using a shared future representation, and (d) DA-WAM, which computes a candidate-conditioned representation for each trajectory and scores the corresponding trajectory–representation pair. The scorer receives the trajectory together with its associated representation and scene context. The panels illustrate conditioning structures rather than an exhaustive classification of prior work.

traffic-rule violations, when distinguishing geometrically similar yet safety-critical trajectories. Realizing this capability requires overcoming two interconnected barriers. At the representation level, predictive features must continue to adapt during planner optimization so that the learned future structure remains aligned with the planning objective. At the planning level, the scorer must evaluate a distinct future latent state for each candidate; shared or pooled futures obscure the action-specific consequences that should guide selection.

To this end, we propose DA-WAM, a decision-aligned world action model. DA-WAM maintains predictive supervision throughout planner optimization via a LoRA-adapted Video Joint Embedding Predictive Architecture (V-JEPA) 2.1 online encoder paired with an exponential moving average (EMA) target encoder, ensuring that future representations co-evolve with the driving task rather than being frozen after pretraining. An action-conditioned predictor generates a distinct future latent state for every trajectory candidate through explicit scene–trajectory interaction, and a multi-head trajectory scorer evaluates each candidate jointly with its corresponding latent state. Because offline logs provide observed futures only for the executed expert trajectory, dense JEPA supervision is applied to the expert-matched candidate, while safety-critical hard negatives supply additional local comparisons near planning boundaries. These hard negatives are geometrically similar to the expert-matched candidate but differ in safety outcomes, discouraging the scorer from relying on geometry alone.

Our contributions are threefold:

- We introduce a compact candidate-conditioned latent-scoring architecture that preserves explicit trajectory–representation pairing while sharing visual scene features across candidates.
- We develop a joint adaptation scheme in which dense future-feature regression is decoded from the expert-matched compact tokens, directly supervising the matched candidate representation and shared visual encoder alongside planning objectives.
- We introduce safety-focused trajectory supervision that sharpens decision boundaries between geometrically similar candidates with different safety outcomes, enabling more reliable driving decisions.

2 Related Work

2.1 Joint-Embedding Prediction and Latent Planning

JEPAs capture high-level semantic and temporal representations by predicting latent features of future states rather than reconstructing pixel-level details Assran et al. (2025); Mur-Labadia et al. (2026). In end-to-end autonomous driving, this paradigm offers an efficient avenue to model scene evolution, which is crucial for forecasting the downstream impact of ego decisions. Recent works such as DriveWorld Min et al. (2024) and LAW Li et al. (2025b) use future dynamics to enrich visual scene representations for motion planning. Drive-JEPA Wang et al. (2026c) adapts pretrained video JEPAs to trajectory planning via fine-tuning, while Auto-JEPA Yang et al. (2026) predicts continuous intent embeddings with a frozen visual backbone to rank candidate paths. Similarly, Latent-WAM Wang et al. (2026b) jointly trains a spatial encoder against an exponential moving average latent target but discards the dynamics branch at test time.

Existing approaches often rely on frozen pretraining, multistage pipelines, or inference-time removal, weakening the coupling between predictive representations and trajectory scoring. In contrast, DA-WAM jointly adapts the shared encoder with predictive and planning objectives, then refines its compact candidate-future branch and scorer for safety-critical discrimination.

2.2 Action-Conditioned Driving World Models

Recognizing that visual environments evolve differently under different ego maneuvers, recent world models explicitly condition future predictions on hypothetical actions or trajectory proposals. Drive-WM Wang et al. (2024) synthesizes multiview video futures under alternative driving commands, whereas LAW Li et al. (2025b) and WoTE Li et al. (2025c) predict trajectory-conditioned latent states or bird's-eye-view (BEV) dynamics. World4Drive Zheng et al. (2025) forecasts multiple intention-guided latent futures and validates paths using an internal evaluator. Concurrently, Drive-Future Hong et al. (2026), IDOL Zhang et al. (2026b), and LCDrive Tan et al. (2026) use future latents for diffusion guidance, inverse-dynamics refinement, and latent chain-of-thought reasoning, respectively.

These methods often aggregate or weakly fuse the predicted futures across proposals, which dilutes the safety consequences specific to each trajectory. In contrast, DA-WAM feeds a distinct compact future representation for each candidate directly into the scorer, establishing an explicit one-to-one correspondence between prediction and action.

2.3 Candidate Generation and Trajectory Scoring

Candidate-based planners generate multiple plausible trajectories and select the optimal one according to scene context and planning objectives. DiffusionDrive efficiently models multimodal action distributions through anchor-guided truncated diffusion Liao et al. (2025). DrivoR compresses multi-camera features into camera-aware register tokens and uses separate transformer decoders to generate and score candidates Kirby et al. (2026). GTRS and ZTRS formulate candidate evaluation as explicit trajectory scoring Li et al. (2025e;d). Beyond geometric and scene-based scoring, DriveSuprim employs coarse-to-fine trajectory selection to distinguish hard-negative trajectories Yao et al. (2026), and BeyondDrive constructs safety-critical, expert-proximate hard negatives to learn safety boundaries in trajectory space Wang et al. (2026a).

Existing candidate scorers remain largely anchored in current scene geometry and rarely incorporate predicted futures on a per-candidate basis. We augment this paradigm with compact action-conditioned future representations to better distinguish geometrically similar candidates with different safety outcomes.

3 Method

Given visual observations X_t and generated trajectory candidates $\mathcal{T} = \{\tau_i\}_{i=1}^{N}$, DA-WAM predicts candidate utilities and make the decision. The model has a shared scene encoder, a compact candidate-conditioned scoring path, and a training-only dense future-feature decoder (Fig. 2). We

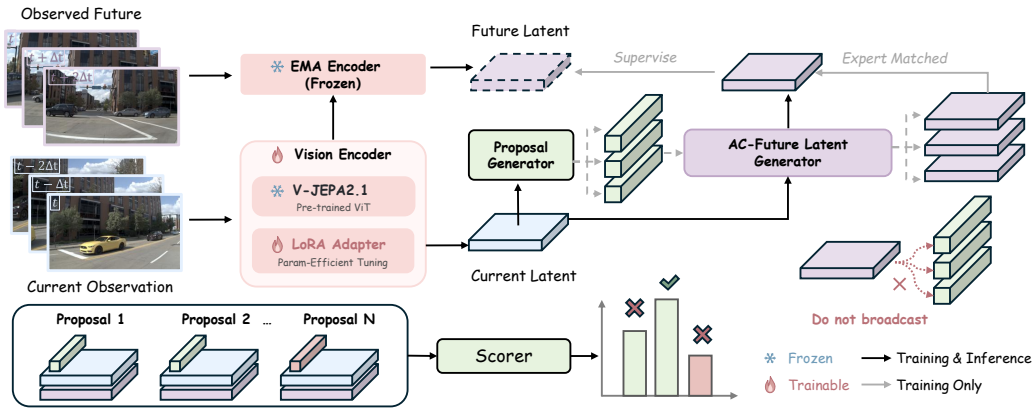

Figure 2: **Overview of DA-WAM.** The online visual encoder produces shared scene tokens. A shared action-conditioned predictor constructs compact tokens for each candidate trajectory, and the scorer evaluates each candidate together with its associated tokens and scene context. A training-only dense decoder and a stop-gradient EMA target provide future-feature supervision for the expert-matched candidate.

describe the inference path first, followed by the predictive and safety-focused supervision used to train it. The candidate count and training settings are reported in Appendix E; a layer-wise specification and consolidated symbol table are provided in Appendices E.6 and E.1.

3.1 Candidate-Conditioned Trajectory Scoring

The visual encoder and trajectory encoder produce scene tokens Z_t and candidate embeddings a_i. A shared compact branch uses each candidate as an attention query over the scene:

$$\begin{aligned} Z_t &= E_\theta(X_t), \qquad a_i = E_\tau(\tau_i), \\ \widehat{z}_i &= P_\phi(Q = a_i, K = Z_t, V = Z_t). \end{aligned} \tag{1}$$

Here $Z_t \in \mathbb{R}^{M\times D}$, and $\widehat{z}_i$ is the compact candidate-conditioned representation. Sharing P_ϕ across candidates makes their representations depend on the trajectory–scene interaction, rather than candidate-specific model parameters.

The scorer combines the current scene, candidate, and associated compact representation:

$$\begin{aligned} h_i &= S_\psi^{\text{enc}}(Z_t, a_i, \widehat{z}_i), \\ \widehat{\mathbf{q}}_i &= S_\psi^{\text{factor}}(h_i), \qquad \widehat{s}_i = S_\psi^{\text{utility}}(h_i), \\ \tau^\star &= \arg\max_{\tau_i \in \mathcal{T}} \widehat{s}_i. \end{aligned} \tag{2}$$

The factor vector contains no-at-fault collision, drivable area compliance, ego progress, time to collision, and comfort. Factor and utility targets are obtained from the planning metrics described in Appendix C. The learned utility, rather than a fixed aggregation of predicted factors, determines the NAVSIM-v1 candidate ranking. The implementation of the shared scorer S_ψ is detailed in Appendix E.6.3.

3.2 Predictive Grounding During Joint Training

We initialize E_θ from V-JEPA 2.1 and adapt its query/value projections with LoRA, keeping the original backbone parameters frozen. An EMA target encoder, updated as $\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta$, encodes the observed future frame with stop-gradient.

Because the log contains only the executed expert future, dense regression is restricted to the closest generated candidate, $i^{\text{exp}} = \arg\min_i \text{ADE}(\tau_i, \tau^{\text{exp}})$. Here ADE is the mean Euclidean (ℓ_2) displacement over planar positions (x, y); yaw is excluded, avoiding any mixing of radians and meters. This nearest candidate is only a proxy for the executed action. The implementation always assigns it the observed future target and does not reject scenes with a large minimum ADE; the remaining

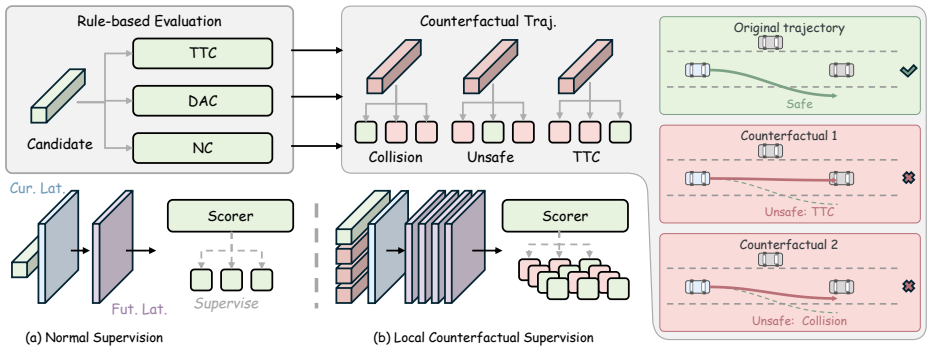

Figure 3: **Safety-focused supervision with retrieved trajectory alternatives.** A planning evaluator assigns utility and safety labels to candidate trajectories. Compared with generated-candidate supervision alone (a), the augmented training set (b) includes expert-proximate alternatives with different collision and time-to-collision outcomes. Generated and retrieved trajectories use the same candidate-conditioned predictor and scorer. Retrieved alternatives receive planning labels, not observed future-feature targets.

candidates receive no direct future-feature target. A dense decoder predicts the matched future token grid:

$$\begin{aligned} \widetilde{Z}_{i^{\mathrm{exp}}} &= D_\omega(Z_t, \widehat{z}_{i^{\mathrm{exp}}}), \qquad Z_{t+\Delta} = \mathrm{sg}(E_{\bar{\theta}}(X_{t+\Delta}))\,, \\ \mathcal{L}_{\mathrm{pred}} &= \frac{1}{MD} \left\| \mathrm{LN}_c(\widetilde{Z}_{i^{\mathrm{exp}}}) - \mathrm{LN}_c(Z_{t+\Delta}) \right\|_1 . \end{aligned} \tag{3}$$

Here LN_c normalizes the channels independently at each spatial position, and the loss is averaged over all M positions and D channels. The dense output $\widetilde{Z}$ is decoded from the expert-matched compact representation $\widehat{z}_{i^{\mathrm{exp}}}$, while only $\widehat{z}_i$ enters the scorer. No stop-gradient is applied between $\widehat{z}_{i^{\mathrm{exp}}}$ and the dense decoder. During joint training, the prediction loss therefore has a direct gradient path into the expert-matched compact tokens, compact predictor, and shared encoder adapters, as well as the proposal generator through its native pose tokens. Planning losses additionally train the compact representations and scorer for all supervised candidates. Thus, direct EMA future-feature supervision applies only to the expert-matched compact representation; other candidates receive planning supervision and share the grounded predictor parameters, but have no observed future target.

3.3 Safety-Focused Supervision and Training

To emphasize local safety differences, we augment generated candidates with external trajectories, including expert-proximate hard negatives. As illustrated in Fig. 3, the hard-negative subset remains geometrically close to the expert while exhibiting worse safety outcomes:

$$d_{\mathrm{traj}}(\tau_j^-, \tau^{\mathrm{exp}}) < \epsilon_{\mathrm{geo}}, \qquad \Delta_{\mathrm{safety}}(\tau_j^-, \tau^{\mathrm{exp}}) > \epsilon_{\mathrm{safety}}. \tag{4}$$

Each retrieved trajectory passes through the same compact branch and scorer and receives planning-derived factor, utility, and preference targets. It is excluded from expert matching and dense future-feature regression because its visual outcome is unobserved. Sampling distributions and thresholds are implementation details, not additional observed-future supervision.

Stage 1: joint adaptation. We jointly train the encoder adapters, dense decoder, compact branch, and planner using

$$\mathcal{L}_1 = \lambda_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{factor}}\mathcal{L}_{\mathrm{factor}} + \lambda_{\mathrm{score}}\mathcal{L}_{\mathrm{score}}. \tag{5}$$

The two planning terms supervise individual driving factors and overall utility, respectively. Generated candidates and external trajectories contribute both targets; the external-trajectory objective is introduced after an initial warm-up. Pairwise safety ranking is not enabled in Stage 1 and is introduced only during Stage 2.

Stage 2: safety-focused refinement. We freeze the visual encoder, including its adapters, the proposal generator, dense decoder, and EMA target, and update only the compact branch and scorer.

The dense future reconstruction is still computed, but its loss weight is set to zero, so it provides no optimization signal during this stage. Stage 2 continues to supervise both the five planning factors and the learned utility for generated and external candidates. External sampling emphasizes safety-critical cases, unsafe-candidate utility losses receive larger weights, and an additional safety-ranking objective emphasizes safe–unsafe preferences. Stage 2 therefore refines candidate discrimination without continuing predictive encoder adaptation.

4 EXPERIMENTS

Benchmarks and Metrics. For NAVSIM, No-at-Fault Collision (NC) measures avoidance of ego-caused collisions, Drivable Area Compliance (DAC) measures whether the trajectory remains within the drivable region, Ego Progress (EP) measures route-aligned advancement, Time to Collision (TTC) captures near-term collision risk, and Comfort evaluates motion smoothness; PDMS aggregates these factors. NAVSIM-v2 additionally reports Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC), which respectively evaluate correct travel direction, red-light compliance, lane adherence, continuity with the observed motion history, and consistency across consecutive predictions; EPDMS is their extended aggregate score. For Bench2Drive, Avg. L2 measures trajectory displacement error, Efficiency reflects driving progress and efficiency, Comfortness measures ride smoothness, Success Rate is the fraction of successfully completed routes, and Driving Score summarizes overall closed-loop driving performance. Full implementation details, dataset protocols, and metric definitions are provided in Appendix C.

Table 1: **NAVSIM-v1 benchmark comparison.**

Method	Venue	NC	DAC	TTC	Comfort	EP	**PDMS**
PDM-Closed Dauner et al. (2023)	CoRL'23	94.6	99.8	89.9	86.9	99.9	89.1
Human driver Dauner et al. (2024)	NeurIPS'24	100.0	100.0	100.0	99.9	87.5	94.8
Ego-stat. MLP Dauner et al. (2024)	NeurIPS'24	93.0	77.3	83.6	100.0	62.8	65.6
UniVLA Wang et al. (2026d)	ICLR'26	96.9	91.1	91.7	96.7	76.8	81.7
VAD-v2 Jiang et al. (2026)	ICLR'26	98.1	94.8	94.3	100.0	80.6	86.2
PRIX Wozniak et al. (2026)	RA-L'26	98.1	96.3	94.1	100.0	82.3	87.8
DIVER Song et al. (2026)	TPAMI'26	98.5	96.5	94.9	100.0	82.6	88.3
DriveVLA-W0 Li et al. (2026)	ICLR'26	98.7	99.1	95.3	99.3	83.3	90.2
ReCogDrive Xiong et al. (2026)	ICLR'26	97.9	97.3	94.9	100.0	87.3	90.8
iPad Guo et al. (2026)	RAL'26	98.6	98.3	94.9	100.0	88.0	91.7
SparseDriveV2 Sun et al. (2026)	ECCV'26	98.5	98.4	95.0	99.9	88.6	92.0
DrivoR Kirby et al. (2026)	CVPR'26	98.9	98.3	96.2	100.0	89.1	93.1
DriveSuprim Yao et al. (2026)	AAAI'26	98.6	98.6	95.5	100.0	91.3	93.5
DA-WAM	–	99.1	98.9	96.8	99.8	90.0	**93.7**

4.1 MAIN RESULTS

Tables 1–3 summarize the main results. DA-WAM achieves 93.7 PDMS on NAVSIM-v1 using its learned candidate scores directly, surpassing prior learned planners while attaining 99.1 NC, 98.9 DAC, and 90.0 EP. Training-free temporal-consistency reranking yields 89.22 EPDMS on NAVSIM-v2, exceeding the strongest comparison by 1.72 points. On Bench2Drive, DA-WAM obtains the highest Driving Score of 59.04 with a 24.66% Success Rate. Notably, while DA-WAM does not lead in progress-oriented metrics, it excels in safety-critical metrics such as NC, TTC, and DAC, highlighting its strength in prioritizing safe planning over mere forward advancement.

4.2 ABLATION STUDIES

Candidate Conditioning and Predictive Supervision. Table 4 isolates the effects of future conditioning and Stage 2 safety-focused supervision. Before Stage 2, the action-conditioned future achieves the highest PDMS of 93.46 among the conditioning variants, outperforming no future prediction (93.31), shared global future (92.81), and current-latent conditioning (93.25). Stage 2 further increases PDMS to 93.68 and improves NC, DAC, TTC, and Comfort, despite a 0.50-point reduction in EP. Fig. 4 tracks the evolution of each metric throughout Stage 2. The overall improvement

Table 2: **NAVSIM-v2 benchmark comparison.**

Method	Img. Backbone	NC	DAC	DDC	TLC	EP	TTC	LK	HC	EC	EPDMS
Ego Status MLP	ResNet-34	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0
TransFuser Chitta et al. (2022)	ResNet-34	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
Hydra-MDP++ Li et al. (2025a)	ResNet-34	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim Yao et al. (2026)	ResNet-34	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS Feng et al. (2026)	ResNet-34	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	98.3	83.1
DiffusionDriveV2 Zou et al. (2025)	ResNet-34	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	87.5
SparseDriveV2 Sun et al. (2026)	ResNet-34	98.1	98.1	99.6	99.8	91.1	97.3	96.9	98.2	78.4	86.7
Hydra-MDP++ Li et al. (2025a)	ViT/L	98.4	98.0	99.4	99.8	87.5	97.7	95.3	98.3	77.4	85.1
DriveSuprim Yao et al. (2026)	ViT/L	97.8	97.9	99.5	99.9	90.6	97.1	96.6	98.3	77.9	86.0
DA-WAM	ViT/L	99.2	98.8	97.8	99.5	91.1	98.7	93.3	96.7	77.1	**89.2**

Table 3: **Bench2Drive closed-loop comparison.**

Method	Avg. L2 (m) ↓	Efficiency ↑	Comfortness ↑	Success Rate (%) ↑	Driving Score ↑
TCP-traj w/o distillation Wu et al. (2022)	1.96	78.78	22.96	20.45	49.30
AD-MLP Zhai et al. (2023)	3.64	48.45	22.63	0.00	18.05
UniAD-Tiny Hu et al. (2023)	0.80	123.92	47.04	13.18	40.73
UniAD-Base Hu et al. (2023)	0.73	129.21	43.58	16.36	45.81
VAD Jiang et al. (2023)	0.91	157.94	46.01	15.00	42.35
GenAD Zheng et al. (2024)	–	–	–	15.90	44.81
EgoFSD-S Su et al. (2024)	0.70	178.30	–	21.00	52.02
ReAL-AD Lu et al. (2025)	0.84	–	–	11.36	41.17
BridgeAD Zhang et al. (2025)	0.71	–	–	22.73	50.06
SeerDrive Zhang et al. (2026a)	0.66	–	–	30.17	58.32
DA-WAM	0.95	140.40	14.64	24.66	**59.04**

Table 4: **Ablation on action-conditioned futures and safety-focused refinement.**

Configuration	Safety sup.	PDMS	NC	DAC	EP	TTC	Comfort
No Future Prediction	✗	93.31	98.45	98.27	91.36	95.48	**99.99**
Shared Global Future	✗	92.81	99.02	98.46	88.68	96.54	**99.99**
Current-Latent Conditioning	✗	93.25	98.44	98.19	**91.38**	95.49	99.94
Action-Conditioned Future	✗	93.46	98.88	98.58	90.47	96.33	99.69
	✓	**93.68**	**99.11**	**98.88**	89.97	**96.81**	99.77

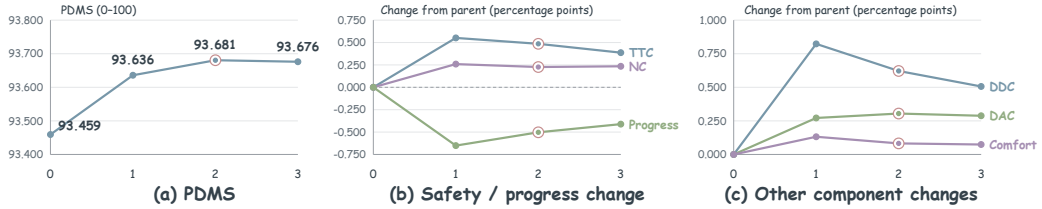

Figure 4: **Safety–progress trade-offs during refinement.** Epoch 0 denotes Stage 1, and epochs 1–3 denote completed refinement epochs; circles mark the retained checkpoint.

is driven primarily by gains in safety-related metrics, while PDMS approaches a plateau after two completed refinement epochs. The improvement indicates that candidate-conditioned future latents provide decision-relevant information for more effective trajectory selection. Table 5 evaluates the predictive-representation design. The V-JEPA 2.1 dense objective improves over the corresponding V-JEPA 2.0 setting for both frozen and LoRA-adapted encoders. With the dense objective enabled, LoRA achieves 92.98 PDMS, outperforming both full fine-tuning (92.62) and the frozen encoder (91.95). Because LoRA keeps the pretrained backbone parameters fixed and updates only low-rank adapters, it preserves the original encoder while adapting its representations to the driving task. Appendix Fig. 8 further shows that LoRA redistributes attention and reorganizes features around decision-relevant vehicles and traffic signals while retaining broader scene context. Finally, under LoRA adaptation with dense supervision, the EMA target achieves the best PDMS of 93.68, supporting the effectiveness of momentum-based target updates.

Sensitivity to Prediction Horizon, Adapter Rank, and Candidate Count. Tables 6–8 study the prediction horizon, LoRA rank, and candidate count, with the first two reporting Stage 1 performance. Each horizon uses a separately trained checkpoint; $\Delta = 0$ predicts the current latent state, while $\Delta = 0.5$ achieves the best PDMS of 93.46 and is used for quantitative evaluation. The 1.5-

Table 5: Ablation of supervision, encoder, and target.

Adaptation	Dense loss	Target	PDMS
Online encoder and predictive objective			
Frozen	✗	Frozen	91.26
Frozen	✓	Frozen	91.95
LoRA	✗	Frozen	92.74
LoRA	✓	Frozen	92.98
Full ft.	✓	Frozen	92.62
Target policy (LoRA + dense loss)			
LoRA	✓	Separate	93.10
LoRA	✓	Shared	93.34
LoRA	✓	**EMA**	**93.68**

Table 6: Ablation on horizon (Stage 1).

Horizon Δ (s)	0.0	**0.5**	1.0	1.5	2.0
PDMS	93.25	**93.46**	93.37	93.23	92.79

Table 7: Ablation on LoRA rank (Stage 1).

LoRA rank r	0	8	16	**32**	64
PDMS	91.95	93.10	93.21	**93.46**	93.08

Table 8: Ablation on candidate count.

Candidates	1	8	16	**32**	64
PDMS	87.11	90.76	91.89	**93.68**	**93.68**

second model is used only for visualization. LoRA rank 32 performs best, while increasing the rank to 64 degrades performance. PDMS increases from 87.11 with one candidate to 93.68 with 32 candidates and remains unchanged with 64, motivating 32 as the default. Appendix B.2 reports the corresponding inference-time scaling.

4.3 Qualitative Analysis

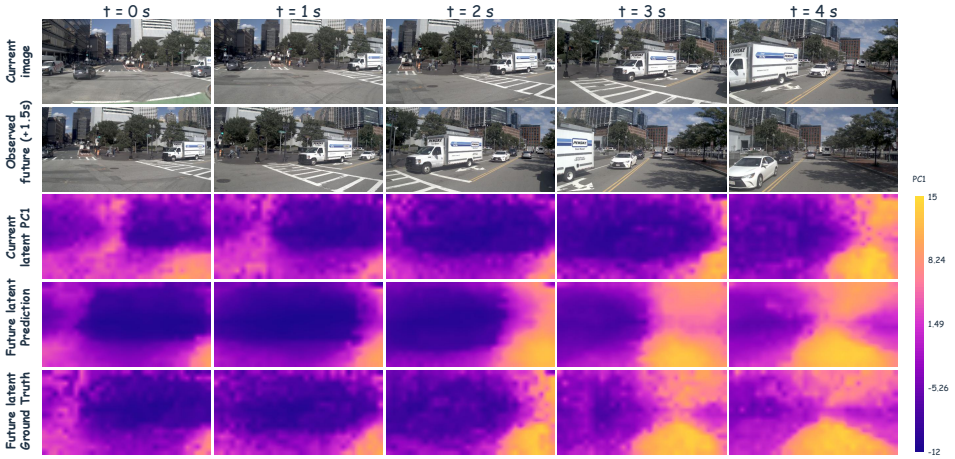

Figure 5: **Future-feature evolution in a truck scene.** Predictions from the 1.5-second visualization checkpoint. Each column shows an independent prediction.

Future-Feature Evolution. Fig. 5 visualizes future-feature prediction in a dynamic truck scene. Although the predictions are smoother and do not recover fine-grained details, they capture the broad spatial evolution needed for decision making rather than simply copying the current representation. In particular, the high-valued yellow regions increasingly align with the open roadway visible in the future frames, suggesting that the framework emphasizes large-scale free-space and scene-layout changes relevant to trajectory evaluation. Additional future-feature visualizations are provided in Appendix Figs. 9 and 10. The PC1 colors represent feature coordinates rather than explicit semantic classes, so this alignment should not be interpreted as a learned drivable-area mask.

Safety-Focused Candidate Refinement. Fig. 6 illustrates the source of the Stage 2 improvement. The first scene features a traffic signal adjacent to a large truck in dense traffic, while the second depicts an intersection signal with multiple interacting vehicles. After refinement, the attention responses become more concentrated on the annotated traffic signals and relevant vehicles. Correspondingly, Stage 2 replaces the collision-prone Stage 1 trajectories with safer alternatives, thereby resolving the NC and TTC failures and increasing PDMS from zero to 1.00 and 0.98, respectively. These examples indicate that safety-focused refinement enhances candidate discrimination by better exploiting safety-critical scene cues.

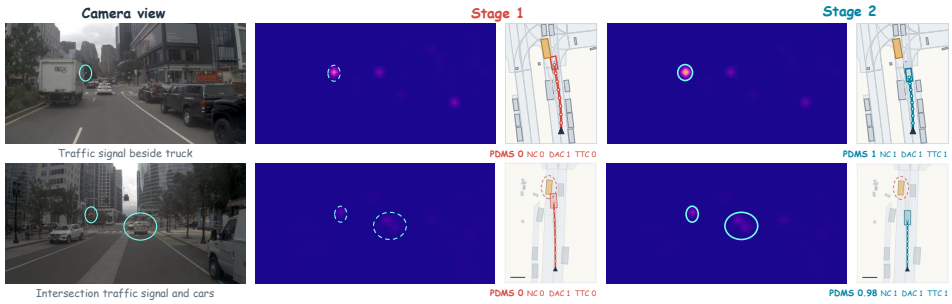

Figure 6: **Safety-critical attention and safer decision making.**



Figure 7: **Direction Following and Safety-Critical Interaction.** Camera views, bird's-eye-view trajectories, and per-scene metrics compare DiffusionDrive, DrivoR, and DA-WAM in a large left turn, tight traffic, and a yielding conflict.

Direction Following and Safety-Critical Interaction. Fig. 7 compares DA-WAM with DiffusionDrive and DrivoR in three representative scenarios. In (a), DA-WAM follows the intended left-turn direction and expert route more closely, achieving higher progress while maintaining safe behavior. In the tight-traffic scene (b), it responds more appropriately to the leading vehicle and avoids the collision risks encountered by both baselines. In the yielding-conflict scene (c), DA-WAM better accounts for the traffic-control cue and crossing conflict, selecting a safe trajectory instead of proceeding into the conflict. Together, these examples show improved direction adherence, interaction with surrounding vehicles, and compliance with traffic controls.

5 Conclusion

We presented DA-WAM, a decision-aligned world action model that integrates compact action-conditioned future representations with candidate-level trajectory scoring. The method combines joint predictive and planning adaptation with safety-focused post-training to align candidate representations with trajectory evaluation. Experiments on NAVSIM-v1, NAVSIM-v2, and Bench2Drive support the approach, with training-free temporal-consistency reranking used on NAVSIM-v2. Ablations favor candidate-specific futures over no-future, current-latent, and shared-future controls, and support LoRA adaptation with dense predictive supervision and an EMA target. Sensitivity studies examine prediction horizon, adapter rank, and candidate count, while qualitative diagnostics characterize representation reorganization and coarse future-feature evolution. Post-training analysis further shows that improved safety and compliance come with a modest reduction in progress. Overall, these findings highlight the value of aligning future representations with candidate evaluation rather than treating prediction as an isolated auxiliary task.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, LaTeX formatting, and code inspection. All technical content, experimental designs, implementations, results, and claims were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The model formulation and two-stage training objectives are described in section 3. Benchmark results and controlled ablations are reported in section 4. Appendix C defines the datasets, evaluation protocols, and metrics, while Appendix E provides the architecture, optimization, post-training, and inference details. Additional representation diagnostics, ablations, and efficiency measurements are provided in Appendices A and B.

REFERENCES

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.

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.

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. doi: 10.1109/LRA.2025.3632616.

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.

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.

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.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. In International Conference on Learning Representations, volume 2026, pp. 68886–68900, 2026.

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, 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, 2026.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, et al. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. arXiv preprint arXiv:2510.24108, 2025d.

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, 2025e.

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.

Yuhang Lu, Jiadong Tu, Yuexin Ma, and Xinge Zhu. Real-ad: Towards human-like reasoning in end-to-end autonomous driving. In 2025 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 27783–27793. IEEE, 2025.

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, and Nicolas Ballas. V-jepa 2.1: Unlocking dense features in video self-supervised learning. In European Conference on Computer Vision, pp. 671–689. Springer, 2026.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2026.

Haisheng Su, Wei Wu, Zhenjie Yang, and Isabel Guan. Egofsd: Ego-centric fully sparse paradigm with uncertainty denoising and iterative refinement for efficient end-to-end self-driving. arXiv preprint arXiv:2409.09777, 2024.

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.

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

Junli Wang, Zhihua Hua, Xueyi Liu, Zebin Xing, Wei Zhang, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, and Qichao Zhang. Beyond imitation: Learning safe end-to-end autonomous driving from hard negatives. In European Conference on Computer Vision, pp. 224–241. Springer, 2026a.

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, 2026b.

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, 2026c.

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, 2024.

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, 2026d.

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.

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.

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.

Jiwei Yang, Zhengxian Chen, Chaosheng Huang, and Jun Li. Auto-jepa: A latent world model of continuous intent for end-to-end autonomous driving. arXiv preprint arXiv:2607.29031, 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.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. arXiv preprint arXiv:2305.10430, 2023.

Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6854–6863. IEEE, 2025.

Bozhou Zhang, Nan Song, Xiatian Zhu, Jiankang Deng, Li Zhang, et al. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. Advances in Neural Information Processing Systems, 38:10204–10229, 2026a.

Chenghao Zhang, Timin Li, and Dongmei Li. Idol: Inverse-dynamics-guided future prediction for end-to-end autonomous driving. arXiv preprint arXiv:2605.31476, 2026b.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In European Conference on Computer Vision, pp. 87–104. 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.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. arXiv preprint arXiv:2512.07745, 2025.

APPENDIX

Table of contents:

- Appendix A analyzes the mechanisms of representation adaptation and dense future-feature prediction.
- Appendix B collects additional ablations and efficiency measurements.
- Appendix C defines the datasets and metrics.
- Appendix D discusses the scope and limitations of the current study.
- Appendix E provides notation, architecture, training, and inference details.

A REPRESENTATION AND FUTURE-PREDICTION ANALYSIS

This section analyzes how Stage 1 LoRA adaptation reshapes the shared visual representation and how the dense JEPA decoder captures future scene evolution. The attention and PC1 visualizations expose complementary aspects of this process: attention reflects changes in the displayed spatial response distribution, whereas PC1 summarizes changes in the organization of the high-dimensional features. The results are descriptive diagnostics of the jointly trained predictive and planning system rather than semantic masks or causal explanations.

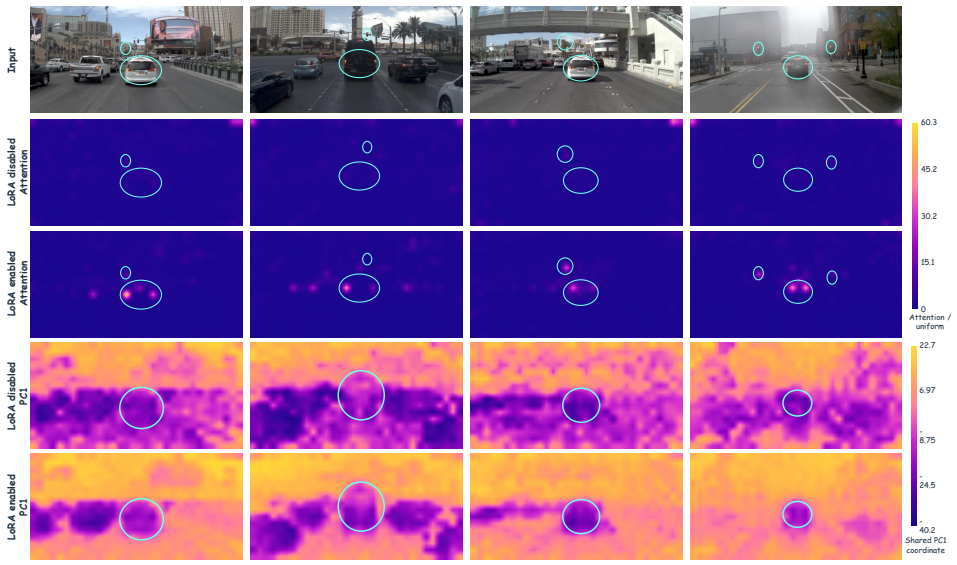

Figure 8: **Spatial reorganization under LoRA adaptation.** Identical input frames are processed with the learned adapters disabled and enabled in the same trained model. Attention maps show a redistribution of spatial response, including clearer local responses near several annotated vehicles and signals; the color scale is relative to uniform attention, not a probability scale. Shared PC1 projections show complementary changes in feature organization while preserving broader scene context.

A.1 SPATIAL REORGANIZATION UNDER LORA ADAPTATION

Fig. 8 compares identical input frames with the learned LoRA adapters disabled and enabled in the same trained model. With adaptation enabled, localized attention responses become more visible around the annotated preceding vehicles and, in some scenes, traffic-signal regions. The shared PC1 projections also change: several foreground regions become more distinct from their surroundings, while the broader spatial context remains visible. The agreement between changes in attention allocation and feature organization is consistent with task-dependent adaptation of the pretrained representation under the joint predictive and planning objectives. However, this adapter-toggle comparison neither separates the contributions of the two objectives nor constitutes a comparison between independently trained models with and without LoRA. The attention color scale reports the

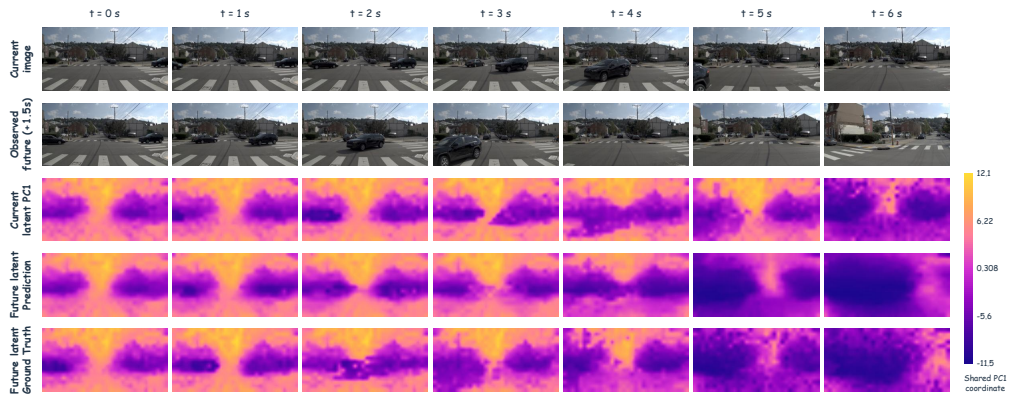

Figure 9: **Future-feature prediction during a crossing interaction.** Each column is an independent 1.5-second-ahead prediction. During the transition, some predicted structures remain closer to the current map than to the target, whereas the $t = 5$ prediction captures part of the target's broader shift toward lower PC1 values. The dense branch therefore follows some coarse future changes but incompletely recovers local and transitional structure.

displayed response relative to uniform attention rather than a raw probability, and PC1 values are feature coordinates rather than semantic labels or measures of causal importance. The plots do not show that information outside the annotated regions has been removed.

A.2 Dense Future-Feature Evolution

Figs. 9 and 10 provide two additional examples from the separately trained 1.5-second visualization checkpoint. This is a single-horizon model distinct from the 0.5-second quantitative checkpoint, rather than the result of horizon conditioning within one model. From top to bottom, the rows show the current image, the observed future image, the current latent, the dense decoder output, and the target future tokens extracted by the EMA encoder. The shared PC1 projection enables spatial comparison between current, predicted, and target representations but does not assign object identities or physical meaning to individual colors. The crossing example emphasizes localized interaction changes, whereas the turning example emphasizes large-scale spatial reorganization accompanying viewpoint changes.

Crossing interaction. In Fig. 9, current, predicted, and target maps share substantial global structure in the earlier columns, so visual similarity alone does not distinguish future prediction from persistence. More informative differences appear during the transition. At $t = 2$ and $t = 4$ seconds, parts of the prediction retain structures that resemble the current map more closely than the corresponding target. At $t = 5$ seconds, by contrast, the prediction reflects the target's shift toward a predominantly lower-valued PC1 layout that is not yet as pronounced in the current features. The dense branch therefore captures part of the coarse future evolution but does not consistently recover local or transitional structure.

Turning sequence. Fig. 10 emphasizes the larger spatial reorganization that accompanies a turn. At $t = 2$ seconds, the current map contains a prominent higher-valued region on the right, whereas both the prediction and target place the dominant higher-valued region on the left. Later columns retain a broadly similar arrangement of higher-valued side regions around a lower-valued central region. These correspondences indicate that the dense predictor does not merely preserve the current feature layout. Agreement is not uniform, however: at $t = 1$ second, the prediction does not fully reproduce the higher-valued structure emerging on the left of the target, and the targets retain finer spatial variation throughout the sequence. This example complements the crossing scene by highlighting viewpoint-dependent feature evolution without assigning semantic identities to the PC1 colors.

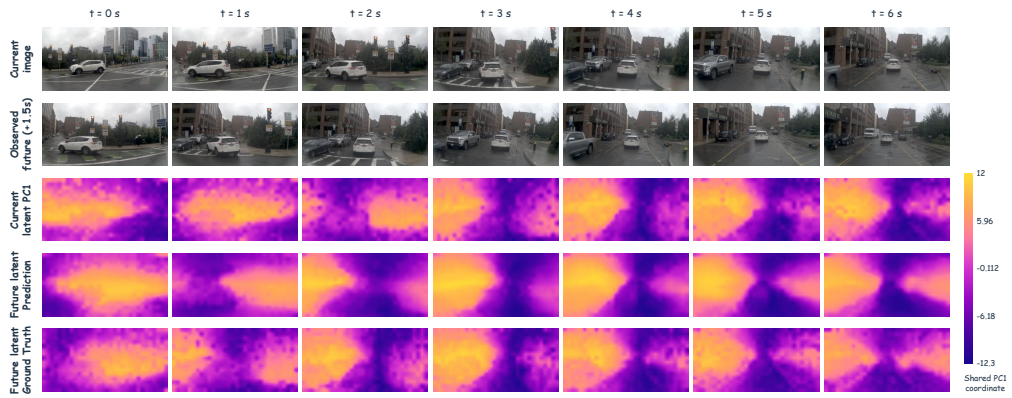

Figure 10: **Future-feature reorganization during a turn.** Each column is an independent 1.5-second-ahead prediction. At $t = 2$ seconds, a dominant higher-valued region shifts from the right in the current map to the left in both prediction and target, illustrating large-scale reorganization rather than fixed persistence. The $t = 1$ transition and finer target structure remain incompletely recovered. PC1 colors are feature coordinates, not object identities or semantic classes.

B Additional Ablations and Efficiency

This section groups supporting analyses that complement the matched ablations in Section 4.2.

B.1 Visual Backbone Ablations

We report the V-JEPA 2.1 backbone comparison in a preliminary planning baseline. These experiments precede the compact candidate-conditioned model and are not backbone replacements retrained within the safety-refined model. All entries use NAVSIM v1 Full Navtest with 12,146 valid scenes. Table 9 compares end-to-end fine-tuning (E2E) and frozen visual encoders.

Table 9: **Exploratory backbone comparison in a preliminary planning baseline.** E2E and Frozen denote end-to-end adaptation and a fixed visual encoder. ViT-g and ViT-G are distinct checkpoint variants. Values are individual-run PDMS scores on a 0–100 scale and do not constitute a controlled scaling study of the final architecture.

Backbone	E2E	Frozen
ViT-B (`vitb`)	86.9958	85.1258
ViT-L (`vitl`)	**88.1473**	**85.5403**
ViT-g (`vitg`)	87.1119	84.6653
ViT-G (`vitG`)	87.9113	81.4653

All runs were configured for 40 training epochs. The E2E runs use a global batch size of 64. The frozen ViT-B/L/g runs also use 64, whereas the frozen ViT-G run uses 128. The ViT-B/L checkpoints are distilled variants, while the ViT-g/G checkpoints follow their respective pretraining recipes. This is therefore a comparison of available backbone configurations rather than a strict parameter-count-only scaling study. ViT-L obtains the highest PDMS in both adaptation settings in these runs. Increasing backbone size does not yield a monotonic improvement, and end-to-end adaptation outperforms freezing for each tested variant.

B.2 Inference Latency and Candidate-Count Scaling

Table 10 compares forward-pass time with DiffusionDrive and DrivoR. For DA-WAM, increasing the candidate count from 1 to 64 keeps time within 36.52–39.00 ms, and 128 candidates require 42.57 ms, indicating effective parallel processing. At the default $N = 32$, DA-WAM takes 37.24 ms: 12.56 ms slower than DiffusionDrive but 258.32 ms faster than DrivoR.

Table 10: **Inference latency and candidate-count scaling.** Reported forward-pass time in milliseconds; lower is better.

Method	Candidates	Forward time (ms) ↓
DiffusionDrive Liao et al. (2025)	–	24.68
DrivoR Kirby et al. (2026)	–	295.56
DA-WAM	1	38.11
	4	36.89
	8	36.52
	16	39.00
	32	37.24
	64	36.97
	128	42.57

C Evaluation Protocol and Metrics

We evaluate on the NAVSIM-v1 and NAVSIM-v2 `navtest` splits. NAVSIM-v1 contains 12,146 driving scenarios Dauner et al. (2024). Unless stated otherwise, all metric values are computed on the native $[0, 1]$ scale and reported after multiplication by 100. As stated in the main text, the DA-WAM result in Table 2 uses the corrected human-filter aggregation, which recomputes the multiplicative and weighted terms after filtering.

NAVSIM-v1 reports No-at-Fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time to Collision (TTC), and Comfort (C). NC and DAC measure collision avoidance and road compliance; TTC indicates whether the trajectory remains free of near-term collision risk; and C checks motion smoothness. EP is derived from the nonnegative start-to-end progress of the simulated ego center projected onto the route centerline and is normalized within the official proposal-scoring procedure using safety-masked progress. It is therefore not the Euclidean or rendered length of the trajectory segment shown in a bird's-eye-view panel; a visually short plotted curve can still receive $\mathrm{EP} = 1$. Safety failures affect the aggregate separately through the multiplicative NC and DAC terms. The per-scenario score is

$$\mathrm{PDMS} = \mathrm{NC}\,\mathrm{DAC}\frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C}}{12}. \tag{6}$$

DDC does not enter NAVSIM-v1 PDMS. When reported in the Stage 2 checkpoint analysis, it is an auxiliary diagnostic rather than a PDMS component.

NAVSIM-v2 extends the evaluation with DDC, Traffic Light Compliance (TLC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). These respectively measure correct driving direction, red-light compliance, lane adherence, continuity with the observed motion history, and consistency across consecutive predictions. Let the tildes denote the official human-reference filtering applied by NAVSIM-v2. Its per-scenario score is

$$\mathrm{EPDMS} = \widetilde{\mathrm{NC}}\,\widetilde{\mathrm{DAC}}\,\widetilde{\mathrm{DDC}}\,\widetilde{\mathrm{TLC}}\frac{5\widetilde{\mathrm{EP}} + 5\widetilde{\mathrm{TTC}} + 2\widetilde{\mathrm{LK}} + 2\widetilde{\mathrm{HC}} + 2\widetilde{\mathrm{EC}}}{16}. \tag{7}$$

D Limitations

First, dense future-feature supervision is available only for the expert-matched candidate because offline logs contain the visual outcome of the executed trajectory but not the counterfactual outcomes of alternative candidates. The remaining candidates receive planning-derived supervision, so their compact future representations should be interpreted as decision-oriented latent consequences rather than reconstructions validated against observed counterfactual videos. Moreover, the dense-feature visualizations evaluate the complete compact-to-dense decoding path; they do not independently establish how much future information is retained in the compact tokens before decoding. The minimum-ADE candidate is also only a proxy for the executed action. The current implementation applies the observed future target without an ADE rejection threshold, and the present experiments do not report the minimum-ADE error distribution. Consequently, the effect of poor expert matches

Table 11: **Notation.** Symbols used in the main formulation and implementation details. Compact candidate-conditioned tokens $\widehat{z}_i$ are distinguished from dense future-feature predictions $\widetilde{Z}_{i^{\text{exp}}}$.

Symbol	Meaning
X_t, $X_{t+\Delta}$	Current and observed future visual inputs.
u_t	Ego-state input used by the trajectory generator.
$\mathcal{T} = \{\tau_i\}_{i=1}^{N}$	Set of N candidate ego trajectories.
τ^{exp}, τ_j^-, $\tau^\star$	Expert, hard-negative, and finally selected trajectories.
Q_i	Native pose-token sequence produced with candidate τ_i.
E_θ, $E_{\bar{\theta}}$	Online encoder and EMA target encoder.
Z_t, $Z_{t+\Delta}$	Current scene tokens and observed future target tokens.
$a_i = E_\tau(\tau_i)$	Candidate query tokens formed from Q_i and the geometry of τ_i.
$\widetilde{Z}_{i^{\text{exp}}} = D_\omega(Z_t, \widehat{z}_{i^{\text{exp}}})$	Dense token prediction decoded from the expert-matched compact tokens for Stage 1 feature alignment.
$\widehat{z}_i = P_\phi(a_i, Z_t)$	Compact candidate-future token sequence used by the scorer.
S_ψ	Shared trajectory scorer conditioned on Z_t, a_i, and $\widehat{z}_i$.
i^{exp}	Index minimizing planar (x, y) ℓ_2 ADE to the expert trajectory; yaw is excluded.
$\mathbf{q}_i$, $\widehat{\mathbf{q}}_i$	Target and predicted planning-factor vectors.
s_i, $\widehat{s}_i$	Target and predicted overall trajectory utilities.
N, T, M, D	Numbers of candidates, trajectory steps, scene tokens, and token dimensions.
μ	Momentum coefficient for the EMA target encoder.
$\mathcal{L}_{\text{pred}}$, $\mathcal{L}_{\text{factor}}$	Future-prediction and planning-factor losses.
$\mathcal{L}_{\text{score}}$, $\mathcal{L}_{\text{rank}}$	Utility-regression and pairwise-ranking losses.
$\mathcal{L}_1$, $\mathcal{L}_2$	Stage 1 joint-adaptation and Stage 2 safety-refinement objectives.
$\lambda_\cdot$	Weights used to combine the training losses.

remains unquantified; future reporting should include this distribution and stratify predictive results by matching error.

Second, most quantitative analysis is based on NAVSIM open-loop evaluation. Bench2Drive provides a complementary closed-loop result, but broader testing across geographic domains, sensor configurations, weather conditions, and rare interactive failures is needed before drawing conclusions about deployment robustness. Likewise, the early backbone comparison consists of individual runs rather than a controlled multi-seed scaling study.

Third, safety-focused post-training changes the operating point rather than improving every metric uniformly: the selected checkpoint gains safety and compliance while sacrificing some progress. The training-free temporal-consistency reranking used on NAVSIM-v2 also introduces an evaluation-time preference for temporal consistency.

E Model and Training Details

This appendix first consolidates the notation and then describes the architecture, training configuration, inference procedure, and layer-wise implementation of the final model.

E.1 Notation

Table 11 summarizes the symbols shared by the main paper and the detailed formulation below.

E.2 Model Architecture

We use a pretrained V-JEPA 2.1 visual encoder with rank-32 LoRA adapters applied to the query and value projections of each Transformer block. The original backbone parameters remain frozen during joint training.

Table 12: **Default architecture and Stage 1 training configuration.** Unless stated otherwise, the model uses 32 generated candidates, rank-32 LoRA, and a 0.5-second predictive target. External candidates are used only for training supervision.

Parameter	Value
Visual encoder	V-JEPA 2.1
LoRA rank	32
LoRA target projections	Query and value
Generated candidates	32
External counterfactual candidates	128
Future prediction horizon	$0.5\,\mathrm{s}$
Future reconstruction weight	0.1
Joint-training epochs	30
Initial learning rate	10^{-4}
Number of GPUs	8
Batch size per GPU	8
Global batch size	64
Mixed precision	bf16

The trajectory generator produces $N = 32$ candidate trajectories, matching the candidate-set notation $\mathcal{T} = \{\tau_i\}_{i=1}^{N}$ used in the main text. For each candidate, a trajectory-conditioned predictor constructs a compact future representation. The shared scorer combines this representation with the candidate trajectory and current scene features to predict a scalar planning score.

During joint training, an exponential moving average (EMA) teacher provides future visual targets from the two-frame window $[t, t + 0.5\,\mathrm{s}]$. A full-latent decoder reconstructs the teacher representation from the ground-truth-matched candidate. The reconstruction objective uses an L1 loss with weight 0.1. The EMA teacher and full-latent reconstruction decoder are used only during training; candidate-conditioned future prediction remains active at inference.

Table 12 describes the checkpoint used for quantitative evaluation. The 1.5-second feature visualizations use a separately trained checkpoint with the same architecture and training setup except for the future target horizon. Each checkpoint is trained for one fixed horizon; the network has no explicit horizon-conditioning input and is not a multi-horizon model. During Stage 1, the proposal generator, LoRA adapters, compact predictor, dense decoder, and scorer are trainable; the pretrained backbone weights remain fixed. Safety refinement is initialized from the Stage 1 checkpoint at epoch 15.

E.3 Joint Training

Using the notation of Eq. 5, the implementation-level decomposition of the Stage 1 objective is

$$\begin{aligned}\mathcal{L}_1^{\mathrm{impl}} = \mathcal{L}_{\mathrm{traj}} + 0.1\,\mathcal{L}_{\mathrm{pred}} + \mathcal{L}_{\mathrm{factor}}^{\mathrm{gen}} + \mathcal{L}_{\mathrm{score}}^{\mathrm{gen}} \\ + \lambda_{\mathrm{ext}}(e)\left(\mathcal{L}_{\mathrm{factor}}^{\mathrm{ext}} + \mathcal{L}_{\mathrm{score}}^{\mathrm{ext}}\right) + \mathcal{L}_{\mathrm{aux}},\end{aligned} \tag{8}$$

where $\mathcal{L}_{\mathrm{pred}}$ is the dense future-feature loss in Eq. 3. For both generated and external candidates, $\mathcal{L}_{\mathrm{factor}}$ applies binary cross-entropy to the five planning-factor outputs and $\mathcal{L}_{\mathrm{score}}$ applies binary cross-entropy to the learned utility output. Both have unit weight within each candidate group. Pairwise ranking is disabled in Stage 1. $\mathcal{L}_{\mathrm{traj}}$ denotes the inherited proposal loss, while $\mathcal{L}_{\mathrm{aux}}$ contains the inherited agent and area auxiliary objectives.

Each scene supplies 128 external counterfactual trajectories, comprising 64 local-hard candidates and 64 balanced candidates. Their coordinates and current-scene features are detached before entering the external scoring branch. Consequently, the external loss updates the candidate-future modules and shared scorer without updating the trajectory generator or visual backbone through this branch.

Using zero-based epoch indices, the counterfactual weight is

$$\lambda_{\mathrm{ext}}(e) = \begin{cases} 0, & 0 \le e \le 4, \\ 0.05(e-4), & 5 \le e \le 9, \\ 0.25, & 10 \le e \le 29. \end{cases} \tag{9}$$

Joint training runs for 30 epochs with an initial learning rate of 10^{-4}, a global batch size of 64, and bf16 mixed precision.

E.4 Safety-Focused Post-Training

We initialize safety-focused post-training from the joint-training checkpoint at epoch 15. We freeze the visual encoder, including its LoRA adapters, the trajectory generator, the full-latent decoder, and the EMA teacher. Only the candidate-conditioned future branch and the shared scorer are optimized. The future reconstruction forward pass is retained, but its loss weight is set to zero during this stage.

Each scene contains 32 generated trajectories and 128 external counterfactual trajectories. The external set consists of 32 local-hard, 32 high-safe, 16 medium-score, 16 collision-failure, 8 drivable-area-failure, and 24 TTC-only-failure candidates.

A candidate is considered safe when

$$\mathrm{NC} \geq 0.95, \qquad \mathrm{DAC} \geq 0.95, \qquad \mathrm{TTC} \geq 0.95. \tag{10}$$

We supervise both the five planning-factor outputs and the learned utility. Both use binary cross-entropy, and the utility-loss contribution of unsafe candidates is multiplied by 4.

For each candidate set, let i^+ denote the safe candidate with the highest target final score, and let $\mathcal{U}_8$ contain unsafe candidates among the eight highest predicted scores. The safety-ranking loss is

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{|\mathcal{U}_8|} \sum_{j \in \mathcal{U}_8} \max(0,\, 0.05 - (\widehat{s}_{i^+} - \widehat{s}_j)) \,, \tag{11}$$

where $\widehat{s}_i$ is the predicted utility from Eq. 2. The loss is zero when no valid safe–unsafe pair exists.

The Stage 2 objective is

$$\begin{aligned} \mathcal{L}_2 = \mathcal{L}^{\mathrm{gen}}_{\mathrm{factor}} + \mathcal{L}^{\mathrm{gen,w}}_{\mathrm{score}} + 0.5 \left(\mathcal{L}^{\mathrm{ext}}_{\mathrm{factor}} + \mathcal{L}^{\mathrm{ext,w}}_{\mathrm{score}}\right) \\ + 0.1 \left(\mathcal{L}^{\mathrm{gen}}_{\mathrm{rank}} + 0.5\, \mathcal{L}^{\mathrm{ext}}_{\mathrm{rank}}\right). \end{aligned} \tag{12}$$

The following table maps the main-text objectives to their implementation-level weights and gradient targets. Here "generated" denotes the 32 proposal-generator candidates and "external" denotes the sampled counterfactual candidates.

Post-training runs for three epochs with a learning rate of 10^{-6} and the same batch size and precision as joint training. The final released model uses the checkpoint after two completed post-training epochs, corresponding to zero-based epoch 1.

E.5 Inference and Temporal Reranking

For NAVSIM v1, inference selects the candidate with the highest learned final score. For the corrected NAVSIM-v2 evaluation, we add training-free temporal-consistency reranking:

$$\widetilde{s}_{t,i} = 0.875\, \widehat{s}_{t,i} + 0.125\, \mathrm{EC}_{\mathrm{pair}} \left(\tau^\star_{t-1}, \tau_{t,i}\right), \tag{13}$$

where $\widehat{s}_{t,i}$ is the learned utility of the current candidate and $\tau^\star_{t-1}$ is the trajectory actually selected at the previous frame. The pairwise consistency term is computed after aligning the trajectories across frames. The selected trajectory is

$$\tau^\star_t = \tau_{t,i^\star}, \qquad i^\star = \arg \max_{i \in \{1,\ldots,32\}} \widetilde{s}_{t,i}. \tag{14}$$

Temporal reranking requires no additional model training. The corrected human filtering, metric computation, and aggregation protocol remain unchanged by reranking.

E.6 Layer-Wise Network Formulation

We formulate the compact candidate-conditioned model used to initialize the safety-refined checkpoint, following the symbols in Section 3.1. Batch dimensions are omitted for clarity. Let X_t denote

Training-loss mapping across the two stages. The Stage 1 external weight $\lambda_{\text{ext}}(e)$ follows the warm-up schedule in Eq. 9.

Loss	Stage 1 weight	Stage 2 weight	Supervision target	Updated modules
Future-feature regression	0.1	0	Expert-matched generated candidate	Stage 1 dense decoder, compact predictor, encoder adapters, and proposal path through native pose tokens
Factor prediction	1.0 generated; $\lambda_{\text{ext}}(e)$ external	1.0 generated; 0.5 external	Five planning factors for all generated and sampled external candidates	Generated Stage 1 path: compact branch, scorer, encoder adapters, and proposal generator through Q_i; external Stage 1 and all Stage 2 paths: compact branch and scorer
Utility prediction	1.0 generated; $\lambda_{\text{ext}}(e)$ external	1.0 generated; 0.5 external; unsafe BCE $\times 4$	Learned-utility target for all generated and sampled external candidates	Generated Stage 1 path: compact branch, scorer, encoder adapters, and proposal generator through Q_i; external Stage 1 and all Stage 2 paths: compact branch and scorer
Safe–unsafe ranking	0	0.1 generated; 0.05 external	Highest-target safe candidate versus unsafe candidates among predicted Top-8	Stage 2 compact branch and scorer
Inherited proposal loss	1.0	0	Generated trajectories against the expert trajectory	Proposal generator and its trainable upstream path
Inherited auxiliary losses	1.0 outer weight	0	Agent and area targets from the base planner	Corresponding base-planner modules

Table 13: **Stage 2 safety-refinement configuration.** The visual encoder, proposal generator, dense decoder, and EMA target are fixed; only the compact predictor and scorer are optimized. Three epochs are run, the model after two completed epochs is retained, and the future-feature loss has zero weight.

Parameter	Value
Learning rate	10^{-6}
Post-training epochs	3
Retained checkpoint	Epoch 1 (zero-based)
Global batch size	64
Generated factor/utility loss weight	1.0
External factor/utility loss weight	0.5
Unsafe utility-BCE multiplier	4.0
Safety-ranking weight	0.1
Safety-ranking Top-K	8
Safety-ranking margin	0.05
Future reconstruction loss weight	0

the available visual observations and u_t the ego state. We write the projected, flattened visual features as the scene tokens

$$Z_t = \text{Flatten}(\Pi(E_\theta(X_t))) \in \mathbb{R}^{M\times D}, \qquad D = 256, \quad M = HW, \quad (H, W) = (16, 32). \tag{15}$$

Thus, the projection Π is absorbed into the abstract encoder notation $Z_t = E_\theta(X_t)$ in the main text. The underlying spatial grid is an image-feature grid, rather than a metric BEV raster. The trajectory generator produces $N = 32$ candidates and their native pose tokens:

$$\big(\{\tau_i\}_{i=1}^N, \{Q_i\}_{i=1}^N\big) = G_\gamma(E_\theta(X_t), u_t), \tag{16}$$

where $\tau_i \in \mathbb{R}^{T\times 3}$ contains (x, y, yaw) coordinates, $Q_i \in \mathbb{R}^{T\times D}$, and $T = 8$. Visual features are computed once and shared across candidates.

E.6.1 TRAJECTORY-CONDITIONED FUTURE TOKENS

Let sg denote stop-gradient and let $[\cdot;\cdot]$ denote channel-wise concatenation. The implementation of the abstract trajectory embedding $a_i = E_\tau(\tau_i)$ and the scene memory are

$$a_i = \psi_f([\mathrm{LN}(Q_i); \eta_f(\mathrm{sg}(\tau_i))]) + E_{\mathrm{pose}}, \tag{17}$$

$$S_t = \mathrm{LN}(Z_t + E_{\mathrm{spatial}}), \tag{18}$$

where η_f is a trajectory MLP and ψ_f is a linear projection followed by LayerNorm and GELU. The future predictor is

$$\widehat{z}_i = P_\phi(a_i, Z_t) = \mathrm{LN}(D_f(a_i, S_t)) \in \mathbb{R}^{T\times D}. \tag{19}$$

Here D_f comprises two pre-normalized Transformer decoder layers with eight attention heads, feed-forward width 1024, and zero dropout. Each layer applies candidate-token self-attention followed by cross-attention to shared scene memory and a feed-forward update. The candidate dimension is folded into the batch dimension during execution. The eight tokens are indexed by trajectory poses; they do not represent eight separately supervised future horizons.

E.6.2 FUTURE RECONSTRUCTION AND GRADIENT BOUNDARIES

During joint training, future reconstruction directly supervises the compact tokens only for the candidate closest to the expert trajectory, denoted by i^{exp} rather than the selection index $i^\star$:

$$i^{\mathrm{exp}} = \arg\min_i \mathrm{ADE}(\tau_i, \tau^{\mathrm{exp}}) = \arg\min_i \frac{1}{T}\sum_{k=1}^{T} \|p_{i,k} - p_k^{\mathrm{exp}}\|_2, \qquad p_{i,k} = (x_{i,k}, y_{i,k}). \tag{20}$$

Although $\tau_{i,k}$ also contains yaw, matching uses only planar position; yaw is not included in the distance. The selected index is detached, and no maximum-ADE rejection threshold is applied. Thus, the observed future target is assigned to the minimum-ADE candidate even when that candidate is not a close approximation of the executed trajectory. A two-layer full-latent decoder maps its compact tokens back to the spatial feature grid:

$$\widetilde{Z}_{i^{\mathrm{exp}}} = D_\omega(Z_t, \widehat{z}_{i^{\mathrm{exp}}}) = \mathrm{LN}[D_{\mathrm{full}}(S_t + E_{\mathrm{full}}, \widehat{z}_{i^{\mathrm{exp}}})] \in \mathbb{R}^{M\times D}. \tag{21}$$

No stop-gradient is applied to $\widehat{z}_{i^{\mathrm{exp}}}$ on this path. Consequently, $\mathcal{L}_{\mathrm{pred}}$ updates the dense decoder, the compact predictor, and the trainable encoder adapters during Stage 1. Because the native pose tokens Q_i are not detached, this loss also has a gradient path into the proposal generator; only the generated trajectory coordinates used by η_f are detached. Compact tokens for candidates other than i^{exp} receive no direct gradient from $\mathcal{L}_{\mathrm{pred}}$. For the quantitative setting $\Delta = 0.5$ seconds, $X_{t+\Delta}$ denotes the observed target window $[I_t, I_{t+0.5}]$. The EMA encoder produces the target tokens

$$Z_{t+\Delta} = \mathrm{sg}(\mathrm{Flatten}(\Pi(E_{\bar\theta}(X_{t+\Delta})))). \tag{22}$$

The implementation instantiates Eq. 3 with per-location channel normalization:

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{MD}\left\|\mathrm{LN}_c(\widetilde{Z}_{i^{\mathrm{exp}}}) - \mathrm{LN}_c(Z_{t+\Delta})\right\|_1, \tag{23}$$

where LN_c normalizes channels independently at each spatial location. The EMA teacher is updated with decay $\mu = 0.99925$:

$$\bar\theta \leftarrow \mu\bar\theta + (1-\mu)\theta. \tag{24}$$

Its parameter set includes the corresponding backbone projection. Future reconstruction has weight 0.1 during joint training and is still computed during safety-focused post-training, where its loss weight is set to zero.

Generated trajectory coordinates are detached before future prediction and scoring, while their native pose tokens retain their computational graph during joint training. Thus, coordinate detachment does not imply that all gradients to the generator are blocked. For external candidates, both coordinates and scene features are detached before the external candidate-future and scoring path. Safety-focused post-training additionally freezes the backbone and trajectory generator parameters explicitly.

E.6.3 CANDIDATE SCORING

The scorer has its own trajectory embedding η_s and fusion mapping ψ_s. It combines native pose tokens, trajectory geometry, and predicted future tokens:

$$U_i = \psi_s([\mathrm{LN}(Q_i); \eta_s(\mathrm{sg}(\tau_i)); \mathrm{LN}(\widehat{z}_i)]) \,, \tag{25}$$

$$h_i = \mathrm{MaxPool}_{k=1}^{T} \left(\mathrm{TransformerEnc}(U_i)\right). \tag{26}$$

The temporal encoder contains one layer. With $V_t = Z_t$, the final feature and multi-head outputs are

$$r_i = \mathrm{LN}(h_i + \mathrm{MHA}(h_i, V_t, V_t)) \,, \tag{27}$$

$$[\widehat{\mathbf{q}}_i; \widehat{s}_i] = \sigma(g_\psi(r_i)) \in \mathbb{R}^6. \tag{28}$$

The first five outputs correspond to the planning-factor vector $\widehat{\mathbf{q}}_i$, and the sixth is the learned utility $\widehat{s}_i$ used for ranking. These are parallel outputs of g_ψ; the predicted factor vector is not fed into the utility output. These layers instantiate the abstract scorer S_ψ in Eq. 2. In the final compact model, the scorer consumes $\widehat{z}_i$ directly; it does not require a full spatial future map for every candidate. The full-latent reconstruction decoder and EMA teacher are omitted from inference.