SiWM: Learning Planning-Relevant World States via Semantic Integration

Anonymous authors
Paper under double-blind review

Abstract

End-to-end autonomous driving often learns planning representations through predefined perception tasks such as object detection, lane structure estimation, and motion prediction. However, it is difficult for a finite set of tasks and labels to cover all scenes and state changes that affect driving decisions on open roads, and this supervision typically requires extensive manual annotation. We quantify perception task losses' impact on closed-loop driving and investigate how to learn environment representations that support it without these losses. We propose the Semantics-Integrated World Model (SiWM), which integrates visual features and continuous planning semantics from a large language model (LLM) into a shared world state, so that both types of information contribute jointly to state learning and recurrent updates. Through action-conditioned future prediction and feature reconstruction, the model learns how the environment evolves and preserves scene information. It generates ego trajectories from the fused state, incorporating driving task information throughout state learning and planning. After 4 task-specific training epochs, SiWM outperforms the direct planning baseline in closed-loop driving. The gains are 20.43 points in Driving Score (DS) and 21.36 percentage points in Success Rate (SR). In some scenarios involving sudden changes, it also achieves better closed-loop performance than ORION, which uses explicit perception task supervision. Further experiments show that integrating planning semantics into world model learning outperforms fusing planning semantics only at the planner, while multi-token representations substantially alleviate collapse from converging state directions across scenes.

1 Introduction

End-to-end autonomous driving directly plans ego trajectories from visual observations (Wu et al., 2022). Training typically includes perception task losses for object detection, lane line estimation, and motion prediction of surrounding traffic participants, providing structured scene information for planning (Hu et al., 2022b; 2023; Jiang et al., 2023).

We analyze their impact on closed-loop planning using ORION, an imitation learning model with explicit perception task losses (Fu et al., 2025). Retaining language and expert ego trajectory supervision, we compare performance before and after removing these losses. Removing these losses leads to a slight decline in the LLM's language output performance (BLEU) and a slight increase in open-loop trajectory error (Avg. L2), whereas the closed-loop Driving Score (DS) and Success Rate (SR) both drop substantially (Figure 1). Open-loop trajectory error measures how well predicted trajectories fit expert trajectories under fixed observations; it cannot, on its own, reflect closed-loop driving ability after plans are executed (Li et al., 2024). This experiment shows perception task supervision's importance for continuous driving and the need to evaluate planning representations' decision-making value through sustained environmental interaction.

However, local states and traffic interactions on open roads are complex and diverse (Prakash et al., 2021). It is difficult for a finite set of perception tasks and labels to cover all changes affecting passage conditions, and their supervision often requires extensive manual annotation.

For example, limited training examples of door-opening events may lead perception models to detect only the vehicle body. Yet the protruding door reduces driving space and may require trajectory

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.

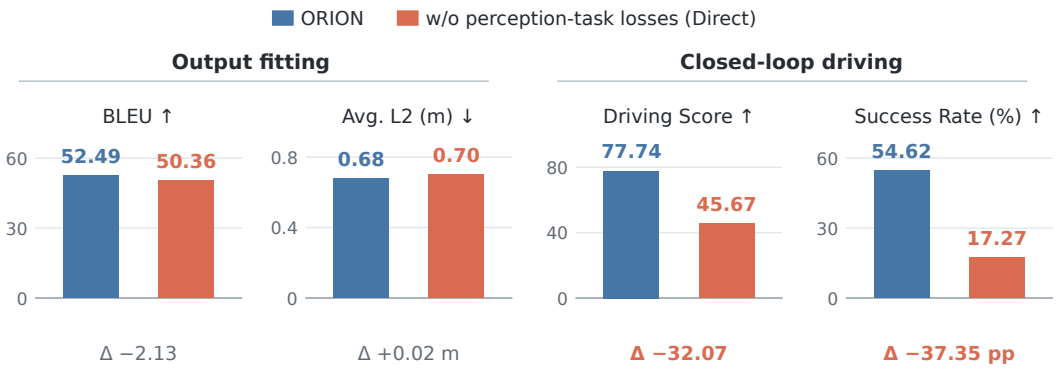


Figure 1: Removing perception task losses from ORION.

adjustments. Planning representations should therefore retain such decision-relevant local states that predefined perception outputs may miss.

Our core question is therefore: **Can a world model use prediction and reconstruction to learn environment representations for planning, without perception-task losses?**

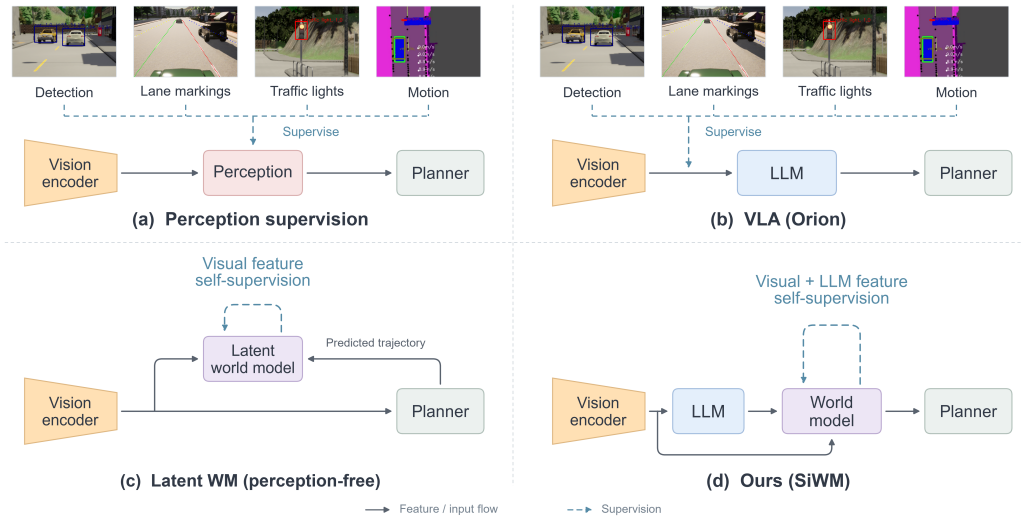


Figure 2: Representative paths for learning planning representations. (a) Perception task supervision; (b) vision-language-action planning with perception supervision (ORION); (c) trajectory-conditioned visual feature prediction (LAW); (d) joint world state learning from visual and LLM features (SiWM).

To investigate this question, we propose the Semantics-Integrated World Model (SiWM), building on ORION's vision-language-action (VLA) framework (Figure 2(b)). As shown in Figure 2(d), SiWM constructs prediction and reconstruction targets from visual and LLM features and uses action-conditioned world state prediction. Through joint recurrent modeling of visual features and planning semantics, it learns world states that support closed-loop driving.

After removing perception task supervision, we learn a world model with feature targets rather than pixel targets. This raises two questions: how to preserve scene differences in the state and how to make the learned information useful for planning, corresponding to state stability and planning relevance.

Regarding state stability, a key challenge is avoiding representation collapse (Chen & He, 2021; Jing et al., 2022), which is also a concern in joint-embedding predictive architectures (JEPAs) (Assran et al., 2023; Bardes et al., 2024). We initially used a recurrent state-space model (RSSM) (Hafner

et al., 2019; 2020) with continuous states to aggregate historical information and predict future states. However, posterior states from different scenes converged toward the same direction, resulting in directional representation collapse. Once persistent directional collapse was observed, continuing the run would no longer provide a meaningful evaluation of the intended recurrent representation-learning design. We therefore did not treat this collapsed run as a completed RSSM baseline for downstream comparison. To address this issue, we represent each recurrent world state with multiple tokens that exchange information through attention, helping preserve scene differences and avoid directional collapse.

Regarding planning relevance, we further incorporate continuous LLM planning features and visual features into a shared world state, jointly representing scene content and its implications for driving decisions. Planning semantics participate in state updates, action-conditioned prediction, and feature reconstruction, linking world state learning to trajectory generation.

After only 4 task-specific training epochs, SiWM achieves DS 66.10 and SR 38.63% on Bench2Drive, exceeding the Direct baseline trained for 18 epochs by 20.43 points and 21.36 percentage points, respectively. Training covers vision-language adaptation, world model and planner training, and joint optimization, totaling 20 hours on 8 NVIDIA H20 GPUs.

With the same data and training epochs, semantic fusion inside the world state (Internal) improves DS by 9.20 points and SR by 6.82 percentage points over fusion at the planner (External). On the five sudden door-opening routes, SiWM achieves mean DS 90.03, exceeding 52.75 for ORION trained with perception task supervision and demonstrating adaptability to changes in local passage conditions (Section 4.2, Table 2).

Our contributions are summarized as follows:

(1) Recurrent world modeling with planning semantics

SiWM integrates continuous LLM planning semantics and visual information into a shared world state for recurrent updates and action-conditioned future prediction. Trajectories are generated from the fused state, incorporating driving task information throughout environment modeling and planning.

(2) Information retention for planning

A multi-token recurrent state uses interactions among visual, semantic, and shared tokens to mitigate directional representation collapse and preserve scene differences. Reconstructing pre-projection visual and LLM features retains both scene content and planning semantics.

(3) Training efficiency and closed-loop gains

SiWM outperforms the 18-epoch Direct baseline in closed-loop driving after only 4 task-specific training epochs. In sudden door-opening tests, it achieves a substantially higher mean DS than perception-supervised ORION, suggesting that SiWM may generalize to long-tail scenarios that conventional perception annotations do not fully capture.

2 Related Work

Latent world models for planning. These methods predict future latent representations to learn environment dynamics, aiming to reduce reliance on perception-task supervision (Li et al., 2025; Zheng et al., 2025b; Wang et al., 2026) (Figure 2(c)). However, accurate prediction does not necessarily lead to effective planning. Retaining decision-relevant information during environment modeling remains an important challenge. **Moreover, improving their adaptability to complex traffic interactions remains an open challenge.**

Vision-language models for driving planning. These methods leverage VLMs' semantic understanding and reasoning capabilities to guide trajectory generation (Shao et al., 2024; Tian et al., 2025; Wang et al., 2025; Li et al., 2026) (Figure 2(b)). However, semantic guidance alone cannot replace learning the scene geometry and dynamics needed for precise planning. Frameworks such as ORION (Fu et al., 2025) learn scene representations through explicit perception-task supervision and use planning semantics to condition trajectory generation. Our ORION ablation (Figure 1) further highlights the role of this supervision in closed-loop driving.

How to retain planning-relevant information while predicting environment dynamics without explicit perception-task losses remains an important question. SiWM integrates visual features and continuous LLM planning semantics into a shared recurrent world state. Through action-conditioned prediction and feature reconstruction, these semantics participate in state learning rather than only conditioning trajectory generation. Trained without perception-task losses, SiWM demonstrates planning capabilities in closed-loop evaluations involving complex traffic interactions. This joint modeling helps retain decision-relevant information that predefined perception annotations may not fully capture.

3 METHOD

SiWM builds on ORION's vision-language feature extraction and conditional VAE planning framework, introducing a recurrent world model between feature extraction and planning (Figure 3). To address these state stability and planning relevance challenges, multi-token recurrent states preserve scene differences and mitigate directional collapse, while planning semantics guide state learning to retain information needed for driving decisions.

3.1 PROBLEM SETUP AND FEATURE REPRESENTATION

The model maps multi-view images I_t, navigation commands, ego-vehicle states, and historical information to future ego trajectories $\hat{\tau}_t$.

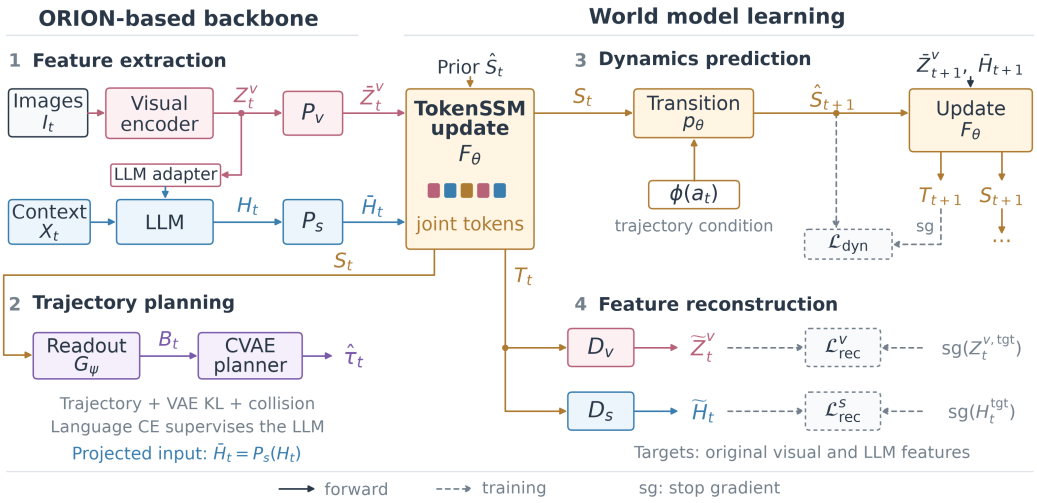


Figure 3: Overview of SiWM. Left: ORION-based feature extraction and planning. Right: recurrent world modeling with prediction and reconstruction. Encoder parameters are shared across time. Solid arrows: forward computation; dashed arrows: training constraints; sg: stop-gradient.

The visual encoder E_v extracts scene features Z_t^v from images; the LLM combines visual information, navigation, and driving context into hidden planning features H_t. Projection modules P_v and P_s map both to a common state dimension for joint state updates:

$$\begin{aligned} Z_t^v &= E_v(I_t), \\ H_t &= R_{\text{plan}}\left[\text{LLM}(V_t^{\text{LLM}}, X_t)\right], \\ \bar{Z}_t^v &= P_v(Z_t^v), \qquad \bar{H}_t = P_s(H_t). \end{aligned} \tag{1}$$

Here, V_t^{LLM} denotes visual features adapted to the LLM input space, X_t denotes navigation and driving context, and R_{plan} reads the LLM hidden state at the special planning token in the planning-QA sequence. $\bar{Z}_t^v$ and $\bar{H}_t$ are the projected visual and semantic features, respectively, for joint world modeling. The two visual input branches use separate feature adapters.

3.2 Recurrent World State Learning

3.2.1 State Stability: Multi-Token Recurrent Modeling

We adopt a recurrent multi-token state structure (Zhai et al., 2025) to preserve differences across scenes and integrate successive observations. We refer to the state modeling module as TokenSSM (token-based state-space model). $S_t \in \mathbb{R}^{K\times d}$ comprises K interacting continuous d-dimensional feature vectors (tokens).

The current visual and semantic features are used to update the prior, yielding the observation posterior state T_t. This posterior is used for reconstruction and as the supervision target for state prediction. Historical posterior states are integrated into the recurrent world state S_t, which is used for planning and for predicting the next state conditioned on the ego trajectory. We denote the recurrent update by F_θ:

$$\begin{aligned}(S_t, T_t) &= F_\theta(\bar{Z}^v_{\le t}, \bar{H}_{\le t}, a_{<t}),\\ \hat{S}_{t+1} &= p_\theta(S_t, \phi(a_t)).\end{aligned} \tag{2}$$

Here, $a_{<t}$ denotes historical action conditions, ϕ encodes ego trajectories into action features, and p_θ is the state transition module. The prior $\hat{S}_t$ predicts the scene before incorporating the current observation, with supervision from the new observation's posterior. History helps distinguish processes such as remaining stationary, approaching, or moving away; trajectory conditioning makes predictions reflect ego motion effects. Training action conditions use expert trajectories; inference conditions combine the current planning output with navigation commands.

At the original 0.1-second frame interval, adjacent features change little, so simply using the current features as the prediction for the next time step can also yield a low prediction error. We set the prediction interval to 1 second so that the same number of steps spans a longer period of evolution. This provides learning signals for accumulated relative displacement and changes in interactions; see Appendix C for sequence organization.

3.2.2 Planning Relevance: Semantic Integration and Feature Reconstruction

To make state learning focus on information needed for driving decisions, we jointly incorporate LLM planning features and visual features into the recurrent state. Visual tokens read visual features, semantic tokens read $\bar{H}_t$, and shared tokens read both inputs. Tokens exchange information through attention (Vaswani et al., 2017), incorporating driving task information into state formation and future prediction. Trajectory supervision constrains the fit to demonstrated actions; feature reconstruction encourages projection and fusion to retain the input information underlying action generation. The visual and semantic decoders reconstruct the original visual and LLM features from the same posterior state. Their outputs have the dimensions of the corresponding features before projection:

$$\widetilde{Z}^v_t = D_v(T_t), \qquad \widetilde{H}_t = D_s(T_t). \tag{3}$$

Both reconstruction errors jointly update the state to retain original visual content and planning semantics.

3.3 Trajectory Planning from the Fused State and Joint Training

The planner aggregates state tokens with learnable queries into the planning condition $B_t = G_\psi(S_t)$. A conditional VAE uses the latent variable z to represent variations in trajectories for a given scene. During training, expert trajectories are used to learn the latent-variable distribution. At inference, the latent variable is drawn from the conditional prior $p_\psi(z \mid B_t)$, and decoder D_ψ uses it together with B_t to generate the trajectory:

$$z \sim p_\psi(z \mid B_t), \qquad \hat{\tau}_t = D_\psi(B_t, z). \tag{4}$$

We jointly optimize world model learning, trajectory planning, and language tasks:

$$\mathcal{L} = \mathcal{L}_{\mathrm{WM}} + \mathcal{L}_{\mathrm{plan}} + \lambda_{\mathrm{lang}}\mathcal{L}_{\mathrm{lang}}. \tag{5}$$

Here, $\mathcal{L}_{\mathrm{WM}}$, $\mathcal{L}_{\mathrm{plan}}$, and $\mathcal{L}_{\mathrm{lang}}$ are the world model, planning, and language losses, respectively. Group definitions include weights, except for the separately specified language coefficient. Appendix A gives the definitions and coefficients of the world model, planning, and language losses.

World Model Objective

$\mathcal{L}_{\text{dyn}}$ matches the action-conditioned predicted prior to the observation posterior state; $\mathcal{L}^{v}_{\text{rec}}$ and $\mathcal{L}^{s}_{\text{rec}}$ supervise original visual and LLM feature reconstruction, respectively, so that learning environmental evolution and retaining scene information jointly constrain the world state.

Planning Objective

The trajectory loss uses only expert trajectories as supervision and measures prediction errors while accounting for navigation modes and valid trajectory points. The KL term measures and constrains the difference between the posterior used during training and the prior used at inference, helping align the inference prior with the latent-variable distribution seen by the decoder during training. The collision loss $\mathcal{L}_{\text{col}}$ penalizes collisions of predicted trajectories with other traffic participants.

Language Objective

$\mathcal{L}_{\text{lang}}$ trains the language branch with autoregressive cross-entropy for visual question answering (VQA) and planning question answering (QA).

4 Experiments

4.1 Experimental Setup

We compare Direct, External, and Internal (SiWM) to study planning representations. Direct uses ORION with only perception-task losses removed. External and Internal retain its feature structures and dimensions, backbone, inputs, training data, evaluation routes, planner type, and language and expert trajectory supervision; they differ in fusing LLM features at the planner or inside the world model, respectively (Figure 4).

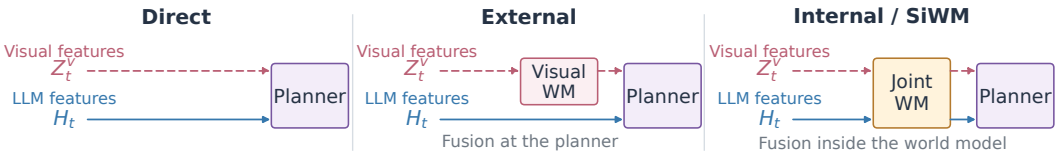


Figure 4: Feature paths for the planning relevance comparison.

4.1.1 System and Training Setup

SiWM completes three stages in 20 hours on 8 NVIDIA H20 GPUs: vision-language adaptation (1 epoch), world model and planner training with fixed upstream vision-language representations (1 epoch), and joint optimization updating language model parameters through LoRA (Hu et al., 2022a) (2 epochs). The model uses pretrained visual representations, signals from the vision-language model and question-answering tasks, and expert ego-vehicle trajectory supervision. World state learning objectives use observations and their subsequent representations. Appendix C provides system parameters and sequence sampling details.

4.1.2 Data and Evaluation Protocol

We evaluate closed-loop driving on the 220 standard Bench2Drive routes (Jia et al., 2024). The sudden door-opening evaluation uses Bench2Drive v0.0.3 route definitions corresponding to ORION's public evaluation, reporting this subset separately. Cross-method comparisons use each paper's reported results.

4.1.3 Evaluation Metrics

Our primary metrics, closed-loop Driving Score (DS) and Success Rate (SR), measure route completion with infraction penalties and the proportion of successfully completed routes, respectively. Open-loop Avg. L2 measures the error between predicted and expert trajectories (in meters), and BLEU (Papineni et al., 2002) measures language output fit.

4.2 Experimental Results

4.2.1 Closed-Loop Planning Performance and Training Cost

In Table 1, SiWM achieves DS 66.10 and SR 38.63% without perception task losses, exceeding Direct by 20.43 points and 21.36 percentage points, respectively, and ranking second among the listed methods on both closed-loop metrics. SiWM has a slightly higher open-loop Avg. L2 than Direct (0.73 versus 0.70 m), but substantially improves closed-loop DS and SR.

Open-loop evaluation compares predicted and expert trajectories using prerecorded observations and does not capture changes in observations resulting from executing a plan. Closed-loop evaluation executes plans and continually updates decisions using new observations, providing a more direct assessment of planning ability during sustained interaction with the environment (Li et al., 2024; Jia et al., 2024). These closed-loop results indicate that joint recurrent learning of visual information and planning semantics can yield environment representations that support sustained driving decisions.

Figure 5 further shows SiWM outperforming Direct in closed-loop evaluation with a shorter task-specific training schedule, demonstrating effective training through world state learning to obtain useful planning representations.

Table 1: Closed-loop driving performance and open-loop trajectory error on Bench2Drive. The top two results for each metric are in bold.

Method	Venue	Explicit perception task losses	VLM supervision	Training	DS↑	SR(%)↑	Avg. L2 (m)↓
UniAD-Base (Hu et al., 2023)	CVPR 23	Yes	No	IL	45.81	16.36	0.73
VAD (Jiang et al., 2023)	ICCV 23	Yes	No	IL	42.35	15.00	0.91
GenAD	ECCV 24	Yes	No	IL	44.81	15.90	–
MomAD	CVPR 25	Yes	No	IL	47.91	18.11	0.82
DriveTransformer-Large (Jia et al., 2025)	ICLR 25	Yes	No	IL	63.46	35.01	**0.62**
ORION	ICCV 25	Yes	Yes	IL	**77.74**	**54.62**	**0.68**
Direct	–	No	Yes	IL	45.67	17.27	0.70
SiWM (ours)	–	No	Yes	IL	**66.10**	**38.63**	0.73

Direct denotes ORION with perception task losses removed. IL denotes imitation learning, and – denotes an unreported result. GenAD and MomAD use publicly reported results (Zheng et al., 2025a; Song et al., 2025).

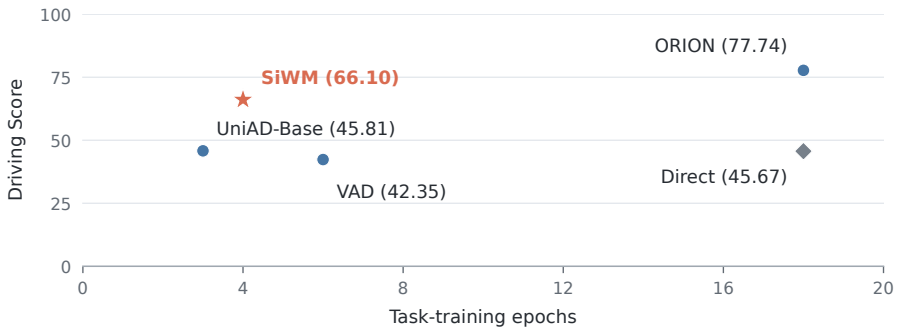


Figure 5: Task-specific training epochs and closed-loop Driving Score, with DS in parentheses.

4.2.2 Adaptation to Local Passage Changes: Sudden Door Opening

To examine the local state changes discussed in the introduction, we evaluate `VehicleOpensDoorTwoWays`: a parked vehicle opens its door as the ego vehicle ap-

proaches, reducing the available driving space and requiring the ego vehicle to avoid it and proceed while accounting for oncoming traffic. Table 2 reports DS on all 5 corresponding standard routes. ORION's results with perception supervision are recomputed from official public route records; SiWM is evaluated on the corresponding routes in this evaluation.

Table 2: Route-level Driving Score in sudden door-opening scenarios. ORION uses perception supervision, while SiWM does not use perception task losses.

Route ID	Map	ORION DS↑	SiWM DS↑
3464	Town13	100.00	100.00
3472	Town13	48.15	**100.00**
3476	Town13	39.13	**64.84**
3482	Town13	24.43	**85.32**
25928	Town11	52.04	**100.00**
Subset mean (5 routes)		52.75	**90.03**

The protruding door changes local shape, occupancy, and the available driving space, while vehicle category and overall motion can remain unchanged; predefined labels centered on categories, overall bounding boxes, and motion have limited expressiveness for these changes. SiWM predicts future observation features, jointly incorporating visual information and planning semantics into recurrent state learning to connect scene changes with the driving task. Results on the five routes show better adaptation to door-opening events, demonstrating the value of world state learning for unusual scene states that are difficult to describe fully with conventional perception labels.

4.2.3 Driving Behavior Visualization

Figure 6 shows planned trajectories during cut-ins, door opening, overtaking cyclists, and interactions with pedestrians, illustrating SiWM's responses to local state changes.

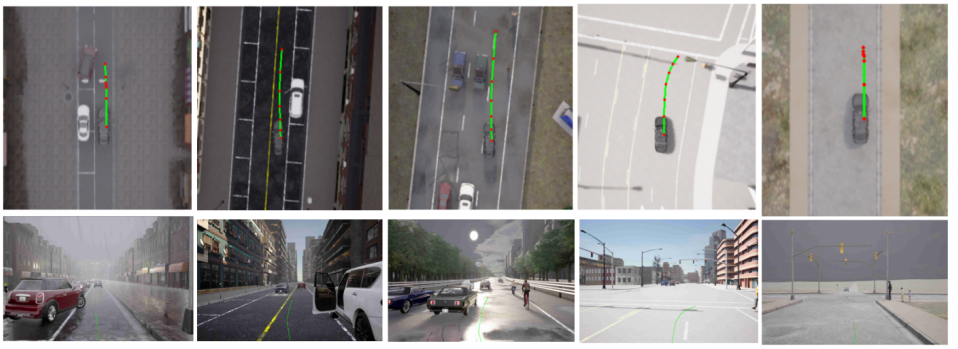


Figure 6: Representative SiWM driving examples. Top: overhead views; bottom: front views. Left to right: cut-in avoidance, door-opening avoidance, overtaking a cyclist, turning right with pedestrian interaction, and pedestrian avoidance.

4.3 Ablation Experiments and State Analysis

4.3.1 State Stability: Multi-Token Representations

We compare cross-scene state similarity in RSSM with continuous states and in multi-token TokenSSM. Directional convergence of states across scenes weakens scene discriminability, a phenomenon called directional collapse. We measure it by mean pairwise cosine similarity across samples: values closer to 1 indicate more consistent directions. See Appendix B for the computation.

In Figure 7(a), RSSM's similarity rises from 0.8741 for visual features to 0.9998 for posterior states, indicating directional convergence. TokenSSM preserves scene differences, with values of 0.4477

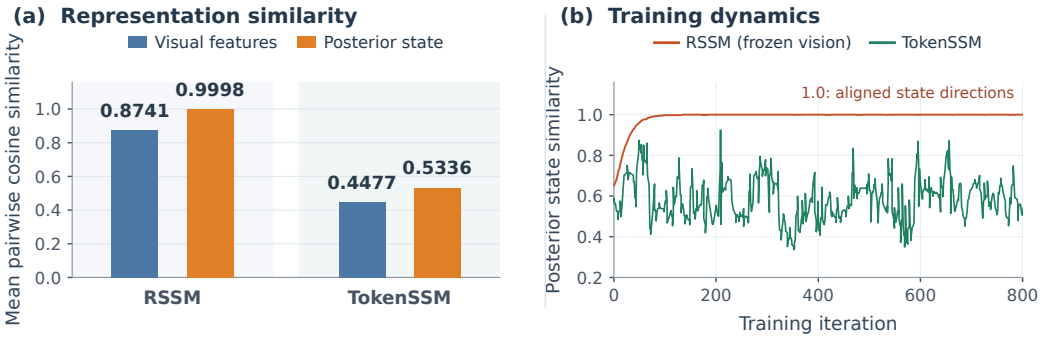


Figure 7: State stability comparison. (a) Mean pairwise cosine similarity of visual features and posterior states; (b) posterior similarity over training iterations. Similarity 1 indicates identical state directions. In (b), RSSM has a frozen visual branch.

and 0.5336, respectively. In (b), RSSM with frozen vision quickly approaches 1.0, while TokenSSM maintains lower similarity. Both diagnostics show that multi-token states effectively mitigate directional collapse and preserve scene differences for joint learning of vision and planning semantics.

4.3.2 Planning Relevance: Internal Semantic Modeling

In Table 3, External achieves DS 56.90 and SR 31.81% through visual world model learning, improving on Direct by 11.23 points and 14.54 percentage points. With the same feature structure and training setup, Internal achieves DS 66.10 and SR 38.63%, gains of 9.20 points and 6.82 percentage points over External. These results support incorporating LLM planning features into world model learning so that driving task information participates in forming and evolving environment representations to better support closed-loop planning.

Table 3: Planning relevance comparison: uses of planning semantics and closed-loop results.

Configuration	Use of LLM features	Training epochs	DS↑	SR(%)↑
Direct	Direct planning, no world model learning	18	45.67	17.27
External	Directly used for planning	4	56.90	31.81
Internal (SiWM)	Internal world model learning and planning	4	**66.10**	**38.63**

5 Discussion and Conclusion

SiWM learns world states for planning by jointly modeling visual information and LLM planning semantics with multi-token recurrent states, action-conditioned future prediction, and feature reconstruction. On the five sudden door-opening routes, SiWM achieves mean DS 90.03, exceeding 52.75 for ORION trained with perception task supervision and demonstrating adaptability to changes in local passage conditions. These results provide experimental evidence that world state learning can support closed-loop driving.

The semantic-path comparison between Internal (SiWM) and External supports joint learning that incorporates planning semantics into world state learning rather than planner-only fusion. State diagnostics support multi-token representations' preservation of differences across scenes. Together, these results support the design principle of preserving environmental differences and using them to inform driving decisions. Semantic integration can help world models learn planning-relevant states for closed-loop driving without perception-task losses.

Ethics Statement

Closed-loop driving evaluation in this work is conducted in the Bench2Drive simulation environment; the sources of evaluation routes and comparison results are described in Section 4.1.2. Before real-world road deployment, further system-level safety validation is needed to assess reliability under different traffic conditions and rare events.

Reproducibility Statement

Section 3 describes the method, and Section 4.1 specifies the experimental setup. Appendices A–D provide details of the training objectives, state similarity computation, system parameters, training sequence organization, and VQA data with question examples. We will open-source the code after publication of this paper.

AI use statement

We used OpenAI Codex to assist with writing, translation, language polishing, figure and table preparation, and typesetting, as well as the retrieval, discovery, and verification of related literature. The authors are responsible for reviewing AI-assisted content and take responsibility for the final paper, experimental results, and citations.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, June 2021.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. ORION: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834, October 2025. doi: 10.1109/ICCV51701.2025.02302.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent – a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022a.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Computer Vision – ECCV 2022*, volume 13698 of *Lecture Notes in Computer Science*, pp. 533–549. Springer Nature Switzerland, 2022b. doi: 10.1007/978-3-031-19839-7_31.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, June 2023. doi: 10.1109/CVPR52729.2023.01712.

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

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, pp. 67227–67243, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

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*, pp. 42942–42959, 2025.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, pp. 157518–157556, 2026.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14864–14873, June 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, June 2021.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130, June 2024.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22432–22441, 2025.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M. Alvarez. OmniDrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22442–22452. IEEE, June 2025. doi: 10.1109/CVPR52734.2025.02090.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6119–6132. Curran Associates, Inc., 2022. doi: 10.52202/068431-0443.

Guangyao Zhai, Xingyuan Zhang, and Nassir Navab. Recurrent world model with tokenized latent states. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. GenAD: Generative end-to-end autonomous driving. In *Computer Vision – ECCV 2024*, volume 15123 of *Lecture Notes in Computer Science*, pp. 87–104. Springer Nature Switzerland, 2025a. doi: 10.1007/978-3-031-73650-6_6.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, XianPeng Lang, and Dongbin Zhao. World4Drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, October 2025b. doi: 10.1109/ICCV51701.2025.02659.

A Training Objectives and Loss Coefficients

Section 3.3 gives the overall training objective and the role of each loss. This appendix specifies their formulas and the coefficients used in the main experiment. Unless otherwise stated, $\langle\cdot\rangle$ denotes the average over valid samples; sg stops gradients through its argument.

A.1 World Model Objective

The world model objective combines dynamics matching and reconstruction in the original visual and LLM feature spaces:

$$\mathcal{L}_{\mathrm{WM}} = \mathcal{L}_{\mathrm{dyn}} + \mathcal{L}^{v}_{\mathrm{rec}} + \mathcal{L}^{s}_{\mathrm{rec}}. \tag{6}$$

Both reconstruction terms use mean squared error (MSE), while dynamics matching uses Smooth L1 with threshold 1. For tensors A and B containing N scalar elements per sample, these distances are

$$\begin{aligned} d_{\mathrm{MSE}}(A,B) &= \frac{1}{N}\sum_{n=1}^{N}(A_n - B_n)^2, \\ d_{\mathrm{SL1}}(A,B) &= \frac{1}{N}\sum_{n=1}^{N} h(A_n - B_n), \qquad h(e) = \begin{cases} \frac{1}{2}e^2, & |e| < 1, \\ |e| - \frac{1}{2}, & |e| \geq 1. \end{cases} \end{aligned} \tag{7}$$

Let $x^{v,\mathrm{tgt}}_u = Z^{v,\mathrm{tgt}}_u$ and $x^{s,\mathrm{tgt}}_u = H^{\mathrm{tgt}}_u$ be the reconstruction targets before projection. At every training time step u, the corresponding observation posterior T_u is decoded into the original feature spaces. With $r \in \{v, s\}$,

$$\begin{aligned} \ell^{r}_{\mathrm{rec},u} &= d_{\mathrm{MSE}}\big(D_r(T_u), \mathrm{sg}(x^{r,\mathrm{tgt}}_u)\big), \\ \mathcal{L}^{r}_{\mathrm{rec}} &= \lambda_{r,0}\langle \ell^{r}_{\mathrm{rec},t}\rangle + \lambda_{r,+}\sum_{k=1}^{J}\rho^k \langle \ell^{r}_{\mathrm{rec},t+k}\rangle. \end{aligned} \tag{8}$$

Here, J denotes the number of future training steps. Current and future reconstructions have separate coefficients, and ρ^k discounts later steps. The dynamics loss matches each predicted prior $\hat{S}_u$ to the posterior obtained from the corresponding observation:

$$\mathcal{L}_{\mathrm{dyn}} = \lambda_{\mathrm{dyn}}\left[\left\langle d_{\mathrm{SL1}}(\hat{S}_t, \mathrm{sg}(T_t))\right\rangle + \sum_{k=1}^{J}\rho^k \left\langle d_{\mathrm{SL1}}(\hat{S}_{t+k}, \mathrm{sg}(T_{t+k}))\right\rangle\right]. \tag{9}$$

Future-step losses are summed with temporal decay rather than normalized by the sum of temporal weights. Expert trajectories provide the action conditions during training.

A.2 Planning Objective

Let $q_t = q_\psi(z \mid B_t, \tau^{\mathrm{gt}}_t)$ and $p_t = p_\psi(z \mid B_t)$ denote the training posterior and inference prior of the conditional VAE (Sohn et al., 2015). The planning objective is

$$\mathcal{L}_{\mathrm{plan}} = \lambda_{\mathrm{traj}}\left\langle \mathbb{E}_{z\sim q_t}\left[\ell^{\mathrm{mask}}_{\mathrm{L1}}(\hat{\tau}_t, \tau^{\mathrm{gt}}_t)\right]\right\rangle + \beta_{\mathrm{KL}}\langle \ell_{\mathrm{KL}}(q_t, p_t)\rangle + \mathcal{L}_{\mathrm{col}}. \tag{10}$$

Trajectory loss. The planner predicts six two-dimensional displacement increments at 0.5-second intervals. Their cumulative sum gives the trajectory over the next 3 seconds. L1 error is computed between predicted and expert increments, with $\Delta\tau_k = \tau_k - \tau_{k-1}$ and $\tau_0 = 0$ in the ego frame. Suppressing the sample and current-time indices, let c_m select the navigation mode m and let v_k indicate a valid trajectory point. The masked loss is

$$\ell^{\mathrm{mask}}_{\mathrm{L1}} = \frac{\sum_{m,k,d} c_m v_k \left|\Delta\hat{\tau}_{mkd} - \Delta\tau^{\mathrm{gt}}_{kd}\right|}{\sum_{m,k,d} c_m v_k}, \qquad k = 1,\ldots,6, \quad d \in \{x, y\}. \tag{11}$$

Samples without valid supervised points are excluded. No additional hard-example weights are used.

VAE KL loss. The conditional VAE represents its latent variable with M_z latent tokens, each with D_z dimensions. For token l, the posterior q_{tl} and prior p_{tl} are diagonal Gaussians with parameters $(\mu^q_{tl}, \sigma^q_{tl})$ and $(\mu^p_{tl}, \sigma^p_{tl})$. The loss sums over dimensions and averages over latent tokens:

$$\begin{aligned}\ell_{\mathrm{KL}}(q_t, p_t) &= \frac{1}{M_z}\sum_{l=1}^{M_z} D_{\mathrm{KL}}(q_{tl}\|p_{tl}) \\ &= \frac{1}{2M_z}\sum_{l=1}^{M_z}\sum_{j=1}^{D_z}\left[2\log\frac{\sigma^p_{tlj}}{\sigma^q_{tlj}} + \frac{(\sigma^q_{tlj})^2 + (\mu^q_{tlj} - \mu^p_{tlj})^2}{(\sigma^p_{tlj})^2} - 1\right].\end{aligned} \tag{12}$$

This term aligns the training posterior with the conditional prior used for trajectory generation at inference.

Collision loss. The distance-threshold penalty checks both trajectory points and the midpoints of consecutive points. Let $e_k = \hat{\tau}_k$ be the cumulative ego position and a_{ik} the position of another traffic participant i, both in the ego frame. Midpoints are $e_{k-1/2} = (e_{k-1} + e_k)/2$ and $a_{i,k-1/2} = (a_{i,k-1} + a_{ik})/2$; the initial positions are used to form the first midpoint. At each checked point r, let $\mathcal{N}_r$ contain valid participants within 3 m of the predicted ego position. For each coordinate d,

$$\begin{aligned}h^d_r &= \left[\delta_d - \min_{i\in\mathcal{N}_r}|e^d_r - a^d_{ir}|\right]_+, \qquad d \in \{x, y\}, \\ g^d_k &= \max(h^d_k, h^d_{k-1/2}), \qquad \mathcal{L}_{\mathrm{col}} = \lambda_{\mathrm{col}}\left\langle \frac{\sum_{k=1}^{6} v_k(g^x_k + g^y_k)}{2\sum_{k=1}^{6} v_k}\right\rangle.\end{aligned} \tag{13}$$

Here, $[b]_+ = \max(b, 0)$, $\delta_x = 1.5$ m, and $\delta_y = 3$ m. If $\mathcal{N}_r$ is empty, $h^d_r = 0$. The point and midpoint penalties are combined by their maximum for each coordinate, and the result is averaged over valid points and both coordinates.

A.3 Language Objective

For VQA and planning QA, let $\mathcal{A}$ contain the supervised token positions (i, j) in a language-branch forward pass, where i indexes samples and j indexes tokens. Given visual inputs V^{LLM}_i and context X_i, the autoregressive cross-entropy is

$$\mathcal{L}_{\mathrm{lang}} = -\frac{1}{|\mathcal{A}|}\sum_{(i,j)\in\mathcal{A}} \log p_{\mathrm{LLM}}(y_{ij} \mid y_{i,<j}, V^{\mathrm{LLM}}_i, X_i). \tag{14}$$

The loss excludes masked prompt and padding positions and assigns equal weight to valid answer tokens.

A.4 Loss Coefficients

Table 4 gives the coefficients for the final training stage used in the main experiment. The world model and planning coefficients are included in their group definitions; the language coefficient multiplies $\mathcal{L}_{\text{lang}}$ in Equation 5.

Table 4: Loss coefficients in the final training stage.

Term	Coefficient	Value
Current visual / LLM reconstruction	$\lambda_{v,0}$ / $\lambda_{s,0}$	3 / 3
Future visual / LLM reconstruction	$\lambda_{v,+}$ / $\lambda_{s,+}$	1.5 / 1.5
Dynamics matching	λ_{dyn}	0.02
Trajectory L1	λ_{traj}	3
VAE KL	β_{KL}	3
Collision	λ_{col}	1
Language cross-entropy	λ_{lang}	1
Future-step decay	ρ	0.9

B Computing State Similarity

We measure directional differences across scenes using the mean pairwise cosine similarity of posterior states within a batch. For each nonzero observation posterior matrix T_i, we flatten it into $z_i = \text{vec}(T_i)$ and normalize it as $\tilde{z}_i = z_i / \|z_i\|_2$. Then,

$$\bar{c} = \frac{2}{B(B-1)} \sum_{i<j} \tilde{z}_i^{\mathsf{T}} \tilde{z}_j. \tag{15}$$

Here, B is the batch size. When $\bar{c} = 1$, state directions are identical across samples, indicating a loss of directional scene discriminability.

C System Parameters and Training Sequence Organization

During training, the visual target branch is updated with an exponential moving average (EMA) (Grill et al., 2020) of the visual branch parameters. TokenSSM uses a 64×256 continuous state comprising 48 visual tokens, 8 semantic tokens, and 8 shared tokens. The observation posterior and prior modules use 4 and 2 attention layers, respectively. Training sequences are organized with ten starting offsets: $0, 10, 20, \ldots$, $1, 11, 21, \ldots$, through $9, 19, 29, \ldots$. Samples within each subsequence are separated by 1 second. Traversing all ten offsets constitutes one full training epoch. This sample interval is distinct from the planner's trajectory point spacing: each output trajectory contains six points at 0.5-second intervals and spans 3 seconds.

D VQA DATA AND QUESTION EXAMPLES

We use the Chat-B2D visual question answering data provided by ORION (Fu et al., 2025). Chat-B2D augments Bench2Drive driving scenes with question-answer annotations covering scene description, critical-object analysis, explanations of ego-vehicle behavior, and recall of historical information. Table 5 lists examples of its public question templates and the information requested, illustrating the content of language supervision.

Table 5: Examples of public ORION/Chat-B2D question templates and their information focus.

Type	Example question	Information requested
Scene description	Can you summarize what the panoramic images show?	Weather, road, and traffic conditions.
Critical-object analysis	Identify the significant objects in the scene and their specific impacts on the ego vehicle.	Important objects and their effects on ego-vehicle behavior.
Ego behavior and reasoning	Please describe your driving behavior and explain the reasons.	Ego actions and reasons based on scene conditions.
Historical information	What was my driving behavior in the previous frame?	The ego vehicle's previous driving behavior.

The question-answer annotations supervise the language branch through the language objective in Section 3.3. The planning feature H_t defined in Section 3 is the LLM hidden state read at the special planning-token position in the planning-QA sequence. It is projected to $\bar{H}_t$ and incorporated into recurrent world state learning, connecting the language branch to the joint modeling of visual information and planning semantics.