RESCENE: VISUALLY MEDIATED VLA ALIGNMENT THROUGH LATENT SCENE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models unify perception, reasoning, and control by mapping visual observations and language instructions to actions. When responding to language-defined situations that are not directly observed, direct language–action supervision alone cannot distinguish visual grounding from an instruction-to-action shortcut. We propose RESCENE, a framework that makes V-L-A alignment through language-specified interventions in visual latent space explicit and testable. RESCENE combines two complementary, shared-weight components: a latent scene editor and a language–action response model. Given observation history, target regions, and an editing condition, the editor masks affected evidence and reconstructs visual features using context, ego motion, and spatial hints. Factual reconstruction anchors these features to observations, while counterfactual construction produces feature-space alternatives for object addition, removal, relocation, and state changes without generating pixels. The response model uses the constructed features as a shared interface for language reasoning and action prediction, without direct access to the edit condition. Response gradients stop at this interface, blocking direct policy-loss updates through latent construction and separating construction from the responses it supports. Reconstruction, language, and action are jointly learned, while normal driving retains a direct observed-feature pathway. On Bench2Drive, RESCENE improves driving score by 6.41 points and route completion by 4.70 percentage points over factual replay, with a higher success rate. Implemented in autonomous driving, RESCENE provides an explicit, visually mediated, and testable formulation of V-L-A alignment for language-defined scene changes. Project page: `https://resceneiclr2027.github.io/`.

1 INTRODUCTION

An autonomous driving agent must connect what it sees, what a task asks, and how it acts. Language provides a common interface for describing traffic, specifying a maneuver, and considering a situation absent from the current observation. Vision-language-action (VLA) models bring these functions into one model, extending vision–language understanding toward control (Zitkovich et al., 2023; Sima et al., 2024; Shao et al., 2024). Their value depends on *alignment*: a description and an action should refer to the same scene and respect its surrounding context.

Existing approaches strengthen language–action alignment through alternative instruction–action pairs (Renz et al., 2025), token-level alignment (Wang et al., 2026), and reasoning tied to trajectory generation (Fu et al., 2025). Counterfactual driving annotations further expand the situations available for supervision (Wang et al., 2025). These advances support responses to language-defined situations, but matching a requested action does not establish how the described change is represented visually. Indeed, low open-loop trajectory error can coexist with weak perception use when ego state is available (Li et al., 2024). The remaining question is how to connect the specified scene to both its description and its action.

Consider the condition *the leading vehicle is absent*. Accelerating may match a training label on a clear road, but a red signal can still require stopping. A model must connect the changed condition with the unchanged traffic context. Direct conditioning can learn this relationship, yet action supervision alone leaves its visual basis ambiguous. Our hypothesis is that *constructing the specified*

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.

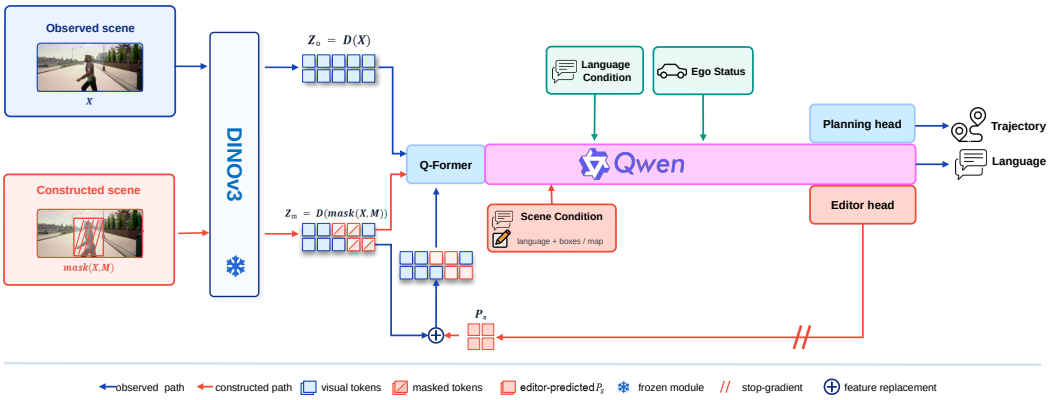


Figure 1: **Visually mediated alignment.** Observed features enter a language–action response model directly. The editor constructs factual or edited features from masked history and a scene condition. A fresh response pass consumes these features after stop-gradient; editor and response model share Q-Former/Qwen weights while keeping their attention contexts separate.

scene in the observed feature space and learning to describe and act from it provides a common basis for both responses.

We introduce **RESCENE**, a joint learning method for *visually mediated alignment*. Given a language condition and spatial hints, a latent scene editor predicts masked features from the remaining observations and ego motion. A language–action response model describes and acts from these features without receiving the edit condition directly (Figure 1). Spatial hints are annotation-derived and used for hypothetical training and offline evaluation.

This use of a constructed scene builds on, but differs from, reconstruction and intermediate reasoning. Masked reconstruction connects feature prediction to language semantics (Yang et al., 2023), and region reconstruction supports visual grounding in manipulation (Song et al., 2026). Intermediate action reasoning can instead use embodied chains of thought (Zawalski et al., 2025) or predicted future image goals (Zhao et al., 2025). RESCENE constructs an alternative to the *observed history*, not a future goal, and uses it for both language and action. Work on confounding and latent interfaces (Zhang et al., 2026; Lin et al., 2026) further motivates distinguishing what a representation encodes from the response it enables.

Our central contribution is a V-L-A learning formulation in which a language-specified scene is constructed in visual feature space and then used as the common input to language and action. Three design choices support this formulation:

- **Construct the specified scene:** predict affected features from masked history and the condition, using factual reconstruction and edited-state constraints where paired edited images are unavailable.
- **Separate scene specification from response:** route the condition through the constructed features and stop response gradients at that interface, keeping construction under visual supervision.
- **Connect hypothetical learning to observed driving:** jointly train observed, reconstructed, and edited views through shared weights while retaining the direct observation pathway for normal driving.

2 RESCENE

RESCENE consists of a latent scene editor and a language–action response model. They share trainable Q-Former/Qwen weights but operate in separate attention contexts.

2.1 Inputs and Conditional Scene Construction

Let X denote camera-indexed observation history and e ego motion. The scene condition $q = (c, g)$ combines language c with actor-box and local-map hints g. It specifies the represented world. A separate task request u and navigation input n specify the response. For example, *the leading vehicle is absent* describes a scene, whereas *predict the route and waypoints* requests an action from that scene.

An annotation-derived mask M covers the original and edited actor regions plus local context. Its support S indexes camera, observed-history time, and patch position. A frozen DINO encoder (Siméoni et al., 2025) produces observed and masked-context features:

$$Z_o = D(X), \quad Z_m = D(\mathrm{mask}(X, M)), \quad P_q = E_\theta(Z_m, e, q, S), \quad Z(q) = \mathcal{A}(Z_m, P_q, S). \tag{1}$$

Assembly replaces support features with P_q and retains Z_m elsewhere. The retained features are masked-context features: encoder contextualization can make them differ from the unmasked Z_o. Masking the RGB before encoding removes direct access to the target pixels during construction.

A shared Q-Former compresses visual context, and Qwen predicts missing DINO features at position queries (Li et al., 2023; Assran et al., 2023; Yang et al., 2025). A factual condition q_f requests the observed scene; an edited condition q_e requests an authored alternative on the same support. ADD, DELETE, and MOVE are applied consistently over actor history and the offline action-authoring horizon. Conditions describe the resulting world, while unedited actors, map context, and observed ego history remain fixed. Construction produces features, not RGB.

2.2 Language and Action from a Shared Visual Interface

The response model receives $V = \mathrm{sg}(Z(q))$, ego motion, task, and navigation. It generates language y and then predicts action a:

$$y \sim p_\theta(\cdot \mid V, e, u, n), \qquad a = F_\theta(V, e, u, n, y). \tag{2}$$

Following SimLingo (Renz et al., 2025), a DrivingAdaptor reads route and waypoint queries after the response tokens. Training uses supervised response tokens; evaluation distinguishes generated-language and fixed-response action prediction. Driving has a fixed request, Commentary and QA describe the world, and Dreamer supplies instruction-following and safety-refusal tasks.

Each response pass starts with a fresh attention context, excluding q, its boxes/map, and editor hidden states and key–value cache. At fixed (e, u, n), the explicit effect of q must pass through V. Stop-gradient blocks response gradients through the construction pass; visual and response losses still update the shared weights through their respective paths. Requests that themselves describe a scene are evaluated separately from fixed-request Driving.

2.3 Visual Supervision

Factual reconstruction has observed image features as targets; authored edits do not. We therefore constrain edited features through supported scene attributes and an inverse return to the observed state.

Factual anchoring. Real images provide the target $T = (Z_o)_S$ for factual prediction P_f. The reconstruction objective combines feature error, direction, and scale:

$$\mathcal{L}_{\mathrm{rec}} = \overline{\ell}_{\mathrm{H}}(P_f, T) + d^{\mathrm{reg}}_{\cos}(P_f, T) + d^{\mathrm{reg}}_{\mathrm{rel}}(P_f, T). \tag{3}$$

Here $\overline{\ell}_{\mathrm{H}}$ is mean SmoothL1 over valid support patches; the two regional terms balance the full support and occupied patches when an object is present.

Edited semantics. For edited features, a reader R_ϕ, trained only on real factual features and then frozen, provides a training constraint on supported scene attributes. For actor units k with valid attribute set H_k and authored targets s_{kh},

$$\mathcal{L}_{\mathrm{sem}} = \mathrm{mean}_k \ \mathrm{mean}_{h \in H_k} \, \ell_h(R_{\phi,h}(Z(q_e); S_k), s_{kh}) \, . \tag{4}$$

Categorical attributes use cross-entropy and position/velocity use SmoothL1. Reader weights remain fixed while gradients reach the editor. Unsupported attributes are masked out.

Inverse feature cycle. To connect the edited feature back to an observed target, the same editor receives an inverse factual condition:

$$P_{e\to f} = E_\theta(Z(q_e), e, q_{e\to f}, S), \qquad \mathcal{L}_{\text{cyc}} = d_{\text{cyc}}(P_{e\to f}, T). \tag{5}$$

After removing the leading vehicle, for example, $q_{e\to f}$ specifies its observed state and boxes so that the editor reconstructs its original DINO support from the edited features. The cycle regularizes reversibility by tying the return from an edited feature to its observed target (Zhu et al., 2017). Appendix A.3 defines the distinct reconstruction and cycle reductions and supported reader heads.

2.4 Three-View Joint Learning

Each group contains an independent observed source i and an authored pair source j. The two constructed views share j and its union mask:

$$V_o^{(i)} = Z_o^{(i)}, \qquad V_r^{(j)} = \text{sg}(Z^{(j)}(q_f)), \qquad V_e^{(j)} = \text{sg}(Z^{(j)}(q_e)). \tag{6}$$

The three views use the same response model. The 1:1:1 ratio counts view occurrences, not distinct sources. Each view receives response-only language cross-entropy and absolute route/waypoint losses in a separate attention context:

$$\mathcal{L}_{\text{resp}}^{(v)} = \mathcal{L}_{\text{lang}}^{(v)} + \mathcal{L}_{\text{route}}^{(v)} + \mathcal{L}_{\text{traj}}^{(v)}, \qquad \mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{cyc}} + \tfrac{1}{3}\sum_{v\in\{o,r,e\}} \mathcal{L}_{\text{resp}}^{(v)}. \tag{7}$$

Default factual Driving targets use Bench2Drive demonstrations (Jia et al., 2024). Edited targets use checked rollouts from a deterministic map-aware planner; an action-preserving edit reuses the factual target only when it remains admissible. Task-specific instructions and refusals use their corresponding authored action references. The action head predicts ten waypoints at 4 Hz over 2.5 s and up to twenty route points at 1-m intervals. Route points beyond observed or executed support are excluded from loss. Appendix A.1 gives target provenance; navigation remains a separate input from these supervised paths.

Training and inference. Scene construction is used during joint learning and hypothetical evaluation. Ordinary driving passes observed features directly to the response model without executing the editor.

3 Experiments

We evaluate transfer to observed driving and language–action responses to authored scene changes on Bench2Drive (Jia et al., 2024). Open-loop driving and language tests use the official validation split; closed-loop tests use the official Bench2Drive220 routes. Appendix B specifies the supported evaluation populations and scoring.

Comparisons. Factual replay is the primary baseline for assessing the benefit of hypothetical scene learning over factual training. Direct conditioning is the primary method-variant control for testing visual mediation against explicit condition access at the response stage. It supplies reconstructed factual features and explicit edit conditions directly to the response, in place of RESCENE's constructed visual features. Its inputs, objectives, and paired results are detailed in Appendix B.3; driving tables also report this control.

3.1 Closed-Loop Evaluation

All methods use observed features and a fixed Driving response, testing closed-loop transfer without online scene editing.

Driving performance. RESCENE raises driving score from 42.69 to 49.10 and route completion from 62.54 to 67.23 relative to factual replay: gains of 6.41 DS and 4.70 percentage points (pp) RC. Success rate also increases, from 12.90% to 17.97% ($+5.07$ pp). RESCENE also has higher DS and RC than direct conditioning; direct conditioning has the highest SR among the three methods. SR counts whole-route successes, while DS and RC also credit partial progress. The bicycle-removal example (Figure 3) illustrates cautious, scene-dependent progress: RESCENE predicts less forward travel than direct conditioning with the cyclist present, then increases it after removal.

Table 1: **Closed-loop driving on Bench2Drive220.** Methods share the same scored-route subset (Appendix B.6). DS: driving score; RC: route completion; SR: success rate. Higher is better.

Method	DS	RC (%)	SR (%)
Factual replay	42.6885	62.5375	12.90
Direct conditioning	45.7248	61.5516	19.82
RESCENE	49.0961	67.2343	17.97

Table 2: **Closed-loop abilities (%).** Official overlapping scenario groups; higher is better. (Appendix B.6).

Method	Overtaking	Merging	Emerg. brake	Give way	Signs	Mean
Factual replay	6.67	11.69	20.00	30.00	6.45	14.96
Direct conditioning	13.33	19.48	28.33	20.00	9.68	18.16
RESCENE	17.78	16.88	23.33	30.00	11.29	19.86

Ability breakdown. RESCENE improves four of the five ability scores over factual replay and matches give-way success, raising the mean from 14.96% to 19.86%. Against direct conditioning, gains in overtaking and give way coexist with lower merging and emergency-braking scores.

3.2 Open-Loop Evaluation

Paired world description and action. For each observed and authored world, the model generates a count answer and then predicts waypoints in the same pass. Both-action pass requires waypoint ADE $\leq$ 1 m in both worlds; both-QA requires both answers correct; joint success requires both conditions.

Table 3: **Generated QA and action on authored validation worlds.** Both-action requires waypoint ADE $\leq$ 1 m in both worlds; both-QA requires both answers correct; joint requires both. The direct-conditioning control is reported in Table 9.

Method	Edited ADE (m) ↓	Both-action (%) ↑	Both-QA (%) ↑	Joint (%) ↑
Factual replay	2.0389	23.68	4.60	0.33
RESCENE	1.3144	42.85	6.69	3.93

Edited-scene action and description. Relative to factual replay, RESCENE reduces edited ADE from 2.0389 to 1.3144 m and raises both-action pass from 23.68% to 42.85%. Both-QA and joint success also improve over factual replay. The direct-conditioning control, including its higher QA and joint success, is reported separately in Appendix B.3.

Instruction-dependent action. With factual visual input and navigation fixed, we compare a neutral request with an instruction and measure proximity to the requested trajectory. Effects vary by request: RESCENE has a larger proximity gain for target speed than direct conditioning, but a smaller gain for relative lane changes. Appendix B.5 reports every request family, including unsupported cases, with the proximity metric defined in the same section.

Table 8 reports factual count, position, motion, and Commentary-action accuracy.

Illustrative response to a rendered edit. Figure 2 shows a selected vehicle-removal case using CARLA re-rendered factual and edited RGB with matched navigation and ego motion. RESCENE's endpoint-change error is 1.16 m, compared with 2.26 m for direct conditioning and 2.73 m for factual replay. Appendix C gives the case-selection protocol and additional mixed and failure cases.



Figure 2: **Vehicle removal in a CARLA re-rendered scene.** The upper panels show factual and edited RGB; the lower panels overlay 2.5-s ego waypoint predictions and the authored action reference in the same ego frame. The green actor future is supplied from the scene record, not predicted by the model; the deleted actor is absent in the edited world. RESCENE approaches the edited action reference more closely in this example. Appendix C details rendering, selection, and physical-admission status.

3.3 Ablation Studies

We compare joint-learning variants on observed driving, examine the editor's reconstruction objective and language conditioning, and test the contribution of constructed features to action prediction.

Observed-driving transfer. Table 4 compares generated waypoint errors on observed scenes against recorded expert trajectories. RESCENE's ADE and FDE are slightly higher than both comparators despite its higher closed-loop DS and RC. We treat expert-trajectory agreement as a secondary diagnostic, with closed-loop performance as the primary measure of driving utility.

Table 4: **Observed-driving transfer across joint-learning variants.** Generated waypoint errors on official Bench2Drive validation, using observed visual input and recorded expert trajectories; lower is better.

Method	Waypoint ADE (m)	Waypoint FDE (m)
Factual replay	0.9874	2.2272
Direct conditioning	0.9763	2.2140
RESCENE	1.0253	2.2688

Reconstruction objective and conditioning. Table 5(a) compares editor training variants. With factual conditions held fixed, adding the regional cosine and relative-error terms reduces object-feature relative L2 by 16.6% over SmoothL1 alone. With the full objective held fixed, concrete factual conditions reduce error by 5.1% relative to a generic reconstruction request. Both effects are consistent across the two matched repetitions.

Table 5: **RESCENE ablations.** (a) Factual editor reconstruction, averaged over two matched training repetitions. (b) Latent substitution with the response model and edited action targets held fixed. The two panels evaluate reconstruction quality and action utility, respectively.

(a) Reconstruction objective and conditioning Editor training	Cosine ↑	Relative L2 ↓
Without regional losses	0.7049	0.7073
Without factual condition	0.7769	0.6216
Full reconstruction setup	**0.8023**	**0.5898**
(b) Latent-path intervention Response input	WP ADE (m) ↓	WP FDE (m) ↓
Factual reconstruction	2.3243	5.5767
Edited reconstruction	**1.6135**	**3.9551**

Latent-path contribution. We keep the response model, navigation, request, and edited action target fixed, replacing only the imagined latent with factual reconstruction on the same mask. Driving ADE increases from 1.6135 to 2.3243 m and FDE from 3.9551 to 5.5767 m. The edited latent therefore improves action fit beyond supplying a reconstructed factual scene. Appendix B.4 details the controlled reconstruction and latent-intervention panels.

4 Limitations

The edited pathway relies on annotation-derived masks and spatial hints; language-only target selection and hypothetical closed-loop deployment are outside this study. Without paired edited images, factual reconstruction, a frozen reader, and inverse consistency constrain but do not uniquely validate the constructed world.

Count QA covers a narrow aspect of scene meaning, and instruction proximity does not establish request completion or refusal safety. Merging and emergency braking remain weaker than direct conditioning, and some collision categories also worsen.

5 Conclusion

RESCENE learns language and action through a shared visual representation of language-specified scene changes. Joint training improves Bench2Drive driving score, route completion, and success rate over factual replay while retaining direct observation-based inference. By separating scene construction from the response it supports, the method provides an explicit, testable formulation of visually mediated V-L-A alignment.

AI Use Statement

Generative AI assisted with literature organization and translation. The authors verified the final manuscript.

Reproducibility Statement

Section 2 defines the method, Appendices A and B its target, interface, and metric contracts, and Section 3 the evaluation protocol. An anonymous release of code, configurations, and evaluation artifacts is not yet prepared.

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*, 2023.

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*, pp. 24823–24834, 2025.

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 Datasets and Benchmarks Track*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742, 2023.

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*, pp. 14864–14873, 2024. doi: 10.1109/CVPR52733.2024.01408.

Jianman Lin, Shailesh Shailesh, Zhongyi Luo, and Jiafei Duan. Breaking the vision-action shortcut: Latent interface training for generalizable robotics foundation models. *arXiv preprint arXiv:2609.12641*, 2026.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11993–12003, 2025.

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*, pp. 15120–15130, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *Proceedings of the European Conference on Computer Vision*, 2024.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. ReconVLA: Reconstructive vision-language-action model as effective robot perceiver. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18549–18557, 2026. doi: 10.1609/aaai.v40i22.38921.

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*, pp. 22442–22452, 2025.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25193–25203, 2026.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shusheng Yang, Yixiao Ge, Kun Yi, Dian Li, Ying Shan, Xiaohu Qie, and Xinggang Wang. RILS: Masked visual reconstruction in language semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23304–23314, 2023.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proceedings of the Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3157–3181, 2025.

Yan Zhang, Yinan Wu, Haoran Duan, and Jungong Han. CofactVLA: Deconfounding vision-language-action models via counterfactual intervention. *arXiv preprint arXiv:2608.04396*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1702–1713, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023.

A Implementation Details

A.1 Data and Action Targets

Edits change an actor's track consistently over observed history and the offline action-authoring horizon. The mask covers original and edited actor regions with a context margin; ADD contributes only its edited region. Unedited actors, map context, and observed ego history remain fixed. Future annotations support action authoring, not observed-history conditioning.

Recorded Bench2Drive demonstrations provide factual Driving targets. Edited targets use the selected map-aware rollout. Action-preserving edits reuse the factual target only when it remains admissible in the edited world. Instructions and refusals use their corresponding authored actions. Waypoints sample the executed path; route supervision ends at the supported distance without extrapolation.

The planner uses a 20-Hz rollout and PDM-Lite-inspired route-and-motion search.[1] Candidates are checked against supplied actor tracks, road occupancy, rules, continuity, and dynamics, then ranked by route progress, jerk, unnecessary lane-change cost, and a deterministic tie-break. These are checks within the authored world, not independent safety ground truth.

A.2 Model and Training

Inputs are front, front-left, and front-right cameras at -1.0, -0.5, and 0 seconds. The model uses frozen DINOv3 ViT-S/16, a shared 64-query, four-layer Q-Former, Qwen3-0.6B with rank-32 LoRA, and a SimLingo-style route/waypoint adaptor. Joint learning starts from factual reconstruction weights with newly initialized driving and target-point adaptors: seed 17, 48 groups/update, 6,550 updates over two epochs, AdamW maximum learning rate $3 \cdot 10^{-5}$, weight decay 0.1, OneCycle warmup 5%, and gradient clip 1.

Each epoch has 157,179 groups, one per distinct factual source. Paired reconstruction/edit sources contribute 83,717 and 83,716 distinct pairs in the two epochs, with one to three exposures per pair. The development schedule has 42,068 factual sources and 22,269 pairs. Observed views cycle Driving, Commentary, and QA; paired views supply Dreamer. Aggregate exposure is approximately 50% Dreamer and one-sixth for each other task, not the same mix within every stream. The 1:1:1 ratio counts views, not distinct sources.

The frozen semantic reader supervises actor type, presence, position, velocity, route blockage, and TTC. Lane availability and signal heads are inactive.

[1] PDM-Lite code and report.

A.3 Loss Definitions and Reductions

Losses reduce within each source before averaging groups, excluding padding. For support S, $p_s, t_s \in \mathbb{R}^{384}$, define $h_s = 384^{-1} \sum_{r=1}^{384} \mathrm{SmoothL1}(p_{sr}, t_{sr})$ with unit transition and $\bar{\ell}_{\mathrm{H}} = |S|^{-1} \sum_s h_s$. For normalized regional weights ω_s,

$$C_\omega(P,T) = \frac{\sum_s \omega_s b_s [1 - \cos_\epsilon(p_s, t_s)]}{\max(\sum_s \omega_s b_s, 10^{-12})}, \qquad b_s = \mathbb{I}[\|t_s\|_2^2 > 10^{-12}], \tag{8}$$

$$Q_\omega(P,T) = \frac{\sum_s \omega_s \|p_s - t_s\|_2^2}{\max(\sum_s \omega_s \|t_s\|_2^2, 10^{-6})}. \tag{9}$$

Cosine uses a 10^{-6} norm floor and clamps to $[-1, 1]$. Uniform weights are $\omega_s^u = 1/|S|$; positive occupied mass gives $\omega_s^o = o_s / \max(\sum_s o_s, 10^{-12})$. Eq. 3 averages uniform and occupied terms: $d_{\cos}^{\mathrm{reg}} = (C_{\omega^u} + C_{\omega^o})/2$ and $d_{\mathrm{rel}}^{\mathrm{reg}} = (Q_{\omega^u} + Q_{\omega^o})/2$. Empty regions use uniform terms only. Occupancy weights the loss, not response input; both coefficients are 1.

Semantic supervision uses categorical cross-entropy and coordinate-mean SmoothL1 on valid active heads, averaged over heads, actor units, then sources. Units with no valid head contribute zero with denominator floor one. Reader parameters are frozen but gradients reach the editor. The inverse cycle is

$$d_{\mathrm{cyc}}(P,T) = \frac{1}{|S|} \sum_{s \in S} \left[h_s + \max(0, 1 - \cos_{10^{-8}}(p_s, t_s)) + \frac{\|p_s - t_s\|_2^2}{\max(\|t_s\|_2^2, 10^{-6})} \right]. \tag{10}$$

It starts from the assembled edited feature without detaching that path. Unlike reconstruction, it averages per-patch relative errors, not a ratio of total errors. Response route/waypoint losses sum coordinate SmoothL1 and average valid points; a fully unsupported route contributes zero. Language CE covers response tokens and one EOS. Response losses average the three views; reconstruction, semantics, and cycle each enter once with unit weight. Full/tail distributed batches normalize by actual global group count.

B Evaluation Details

B.1 Comparisons and Evaluation Scope

Table 6 summarizes the hypothetical-response inputs and training objectives of RESCENE, the factual replay baseline, and the direct-conditioning control. Direct conditions supply selected-actor cues rather than the full QA answer or an action target.

Table 6: **Primary baseline and direct-conditioning ablation.** All methods retain factual reconstruction and response supervision. The final column specifies additional edited-feature supervision.

Method	Response visual input	Direct edit cues	Edited-feature losses
Factual replay	Factual reconstruction	None	None
Direct conditioning	Factual reconstruction	Text and geometry	None
RESCENE	Constructed edited features	None	Semantics and cycle

Open-loop evaluation uses Bench2Drive validation split. Table 7 summarizes the evaluation populations.

B.2 Generated Language and Action

Waypoint ADE averages ten 4-Hz Euclidean errors; FDE uses the final point. Paired count QA is generated before action prediction. Both-QA requires correct answers in both worlds, both-action requires both ADEs ≤ 1 m, and joint success requires both. Factual targets are recorded expert paths; edited targets are authored planner paths.

Factual-language evaluation scores supported count, nearest-position, observed-motion, and Commentary-action slots; unparsed answers are incorrect. Table 8 reports accuracy for each slot.

Table 7: **Evaluation scope.** All comparisons within a panel use matched supported examples. Closed-loop uses the shared scored-route subset.

Panel	Population	Response mode
Observed Driving	Val: 11,058 sources / 50 routes	Generated
Authored QA/action	Val subset: 1,195 pairs / 30 routes	Generated
Instruction	Val subset: 417 requests / 131 sources	Generated
Closed-loop	B2D220: 217 scored routes	Fixed Driving

Table 8: **Generated factual language on supported validation slots.** Conservative slot accuracy, including unparsed responses as incorrect; higher is better.

Method	QA count	QA nearest	QA motion	Commentary action
Factual replay	0.421	0.793	0.319	0.477
Direct conditioning	0.417	0.786	0.297	0.470
RESCENE	0.407	0.799	0.313	0.464

Aggregation. Main-table means weight sources equally. The supplementary request and subset effects weight routes equally, then sources within each route.

B.3 Direct-Conditioning Control

The direct-conditioning variant replaces edited features with factual reconstruction and explicit edit text/geometry. It retains factual reconstruction and response supervision but omits edited-feature semantic and cycle losses (Table 6). Under the Table 3 protocol, RESCENE has lower edited ADE and higher both-action pass; direct conditioning has higher both-QA and joint success (Table 9).

Table 9: **Direct-conditioning ablation on authored validation worlds.** Same protocol as Table 3; inputs and objectives are specified in Table 6.

Method	Edited ADE (m) ↓	Both-action (%) ↑	Both-QA (%) ↑	Joint (%) ↑
Direct conditioning	1.3489	36.15	14.56	6.03
RESCENE	1.3144	42.85	6.69	3.93

Table 10 breaks down the comparison by authored-world panel.

Table 10: **Sensitivity to the authored-world panel.** Equal-route/source RESCENE-minus-direct effects in percentage points. Count invariance does not imply action invariance.

Panel	Both-QA difference	Joint difference
All supported	−0.49	−1.61
Stricter subset	−7.31	−0.13
Changed counts	−1.15	−1.30
Unchanged counts	+6.20	−1.75

B.4 Editor and Latent-Path Ablations

Table 5 separates an editor-pretraining comparison from an inference-time intervention on a fixed trained response model. These controlled panels are separate from the official-validation driving and language benchmarks.

Reconstruction comparison. Each variant uses the same 2,048 training sources, 512 held-out sources, initialization and data order within each repetition, and 516 updates. The full setup combines SmoothL1, regional cosine and relative-error losses with concrete factual conditions. Removing the regional terms retains the conditions; removing factual conditions retains the full loss but

uses a generic reconstruction request. Language supervision is unchanged. Cosine and relative L2 are measured on object-class patches within the target boxes, using 384 object sources; empty regions are excluded from this object metric. The table averages source means across two repetitions. Relative L2 in the two runs is $0.7096/0.7050$ without regional losses, $0.6224/0.6209$ without factual conditions, and $0.5900/0.5896$ with both.

Latent intervention. For each source, the same mask is reconstructed under either its edited or factual condition. Only the latent entering the edited-world response pass changes: the checkpoint, navigation, task prompt, and edited action reference remain fixed. The table reports autoregressive Driving; its fixed-response action scores agree. Predictions on the untouched observed and factual-reconstruction views are identical across the two executions. The reported 0.7108 m ADE increase is the paired effect of substituting factual for edited features, rather than a comparison between separately trained response models.

B.5 Instruction Evaluation

The same factual visual and navigation inputs accompany neutral and requested instructions. Proximity gain is $\mathrm{ADE}(\widehat{A}_{\mathrm{neutral}}, Y_{\mathrm{request}}) - \mathrm{ADE}(\widehat{A}_{\mathrm{request}}, Y_{\mathrm{request}})$. Table 11 reports the RESCENE-minus-direct difference by request family and marks unsupported or insufficiently represented cases. This panel measures neutral-to-requested trajectory proximity; request completion, safety refusal, and shuffled-instruction sensitivity require separate evaluations.

Table 11: **Instruction-dependent proximity gain.** RESCENE-minus-direct effects in meters; positive favors RESCENE. All request families are shown. Proximity gain measures the reduction in requested-trajectory ADE from the neutral to the instructed prediction.

Request family	Proximity-gain difference (m)
Collision	$+0.087$
Location-specific collision	$+0.383$
Drive over an object	Unsupported
Increase speed	$+0.156$
Faster collision	$+1.134$
Lane-change transition	-0.157
Absolute lane change	-0.024
Absolute lane change (all lanes)	-0.081
Relative lane change	-0.263
Parking	$+0.176$
Red-light violation	Unsupported
Sidewalk driving	$+0.143$
Decrease speed	-0.009
Immediate stop	$+0.110$
Target speed	$+0.174$
Pedestrian-directed request	Insufficient support

B.6 Closed-Loop Scoring

We use official Bench2Drive220 routes and per-route scoring. Comparisons use the same 217 scored routes; three routes are excluded for repeatable CARLA failures. DS and RC average those route scores. SR counts completed routes without infractions other than minimum-speed infractions.

Inference uses observed features, a fixed Driving response, and sparse near/far navigation without scene construction. Ability groups overlap and score whole-route success. Traffic Signs additionally requires junction passage without red-light/stop infractions.

C QUALITATIVE CASES

Figure 2 and the two cases below use CARLA re-rendered RGB, matched navigation and ego motion, a fixed `Waypoints:` response, and 2.5-s waypoint predictions. The figures evaluate responses to rendered inputs. Actor futures and action references are supplied by the scene record and authoring process; the ego waypoint predictions are model outputs. Evaluation of constructed latent semantics is a separate protocol.



Figure 3: **Bicycle removal: cautious progress with mixed accuracy.** RESCENE predicts less forward travel than direct conditioning with the cyclist present, then increases it after removal. It reduces endpoint-change error relative to direct conditioning (2.76 versus 4.91 m), but its edited-world FDE is higher (6.02 versus 5.44 m). Factual and edited RGB are CARLA re-renderings; the green line is the supplied target-actor future.



Figure 4: **Bicycle relocation: unnecessary action change.** The authored factual and edited action references are identical, yet all three methods change their actions. RESCENE's endpoint-change error is 7.56 m, versus 9.44 m for direct conditioning and 6.09 m for factual replay. The green paths show supplied actor futures, not model-predicted motion.