RESTORING SPATIAL DISTANCE FEEDBACK IN TOKEN-NATIVE DRIVING VLAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discretizing continuous trajectory waypoints directly into a Vision Language Action (VLA) model's vocabulary allows it to treat motor control as standard autoregressive generation. This unified formulation ensures that policy execution directly inherits the pre-training capabilities and scaling law benefits of the underlying Vision Language Model (VLM), bypassing the need for specialized continuous regression heads. However, optimizing these tokenized action spaces with standard categorical cross-entropy imposes an *isometric loss landscape* that penalizes all incorrect tokens uniformly, regardless of their physical error in metric space. We demonstrate that this loss geometry causes closed-loop policy failure in models trained with discrete K-means tokens (K-DISK) despite high action reconstruction accuracy. We address this bottleneck through two core contributions: 1) adapting Ordered Action Tokenization (OAT) from robotics to autonomous driving, where continuous trajectory proximity is mapped to shared sequence prefixes, and 2) Coarse Trajectory Anchor Regularization (CTAR), a differentiable metric loss that backpropagates smooth-L_1 trajectory errors directly through the first coarse token via a frozen decoder. This provides the explicit metric feedback missing from pure cross-entropy. Evaluated on CARLA Bench2Drive and on NAVSIM v2, our method yields significant gains over standard K-DISK (+33.57 DS, +0.0979 EPDMS) and matches the performance continuous action heads (+1.55 DS, +0.0425 EPDMS), demonstrating that VLAs can achieve closed-loop fidelity without departing from native token architectures.

1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a dominant paradigm for end-to-end driving and generalist embodied decision-making (Wang et al., 2026; Zhou et al., 2026; Wu et al., 2025; Chen et al., 2024; Kim et al., 2024). Unifying perception, natural language reasoning, and motor control directly within a single architecture enables effective multi-modal reasoning (Ye et al., 2026) and compatibility with optimized LLM serving infrastructure, such as KV-caching (Kwon et al., 2023), speculative decoding (Leviathan et al., 2023), and quantization (Liu et al., 2025). Furthermore, as frontier multimodal systems exhibit increasingly impressive spatial reasoning in embodied domains (OpenAI, 2026; Google DeepMind, 2025), maintaining a pure token-native action stream ensures policies directly inherit these foundation-scale pre-training and scaling law benefits.

Despite these advantages, achieving robust closed-loop driving solely through discrete action tokens remains a persistent challenge. To circumvent performance degradation from discretization, recent state-of-the-art VLAs rely on architectural modifications, such as attaching specialized continuous regression heads (Renz et al., 2025; Shao et al., 2024; Fu et al., 2026; Liu et al., 2026b) or implementing complex hybrid pipelines that pre-train on discrete tokens but transition to continuous diffusion decoders for control execution (Wang et al., 2026). By bypassing discrete action spaces, these continuous modifications allow models to output smooth, fine-grained trajectories required for closed-loop stability.

In this work, we show that the closed-loop failure of token-native VLAs stems from an optimization bottleneck rather than an inherent constraint of tokenization. When continuous action trajectories are clustered into unstructured codebooks (e.g., K-means centroids (Zhou et al., 2026; Xiong et al., 2026)), training under standard categorical cross-entropy ($\mathcal{L}_{\text{CE}}$) induces an *isometric* loss landscape.

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.

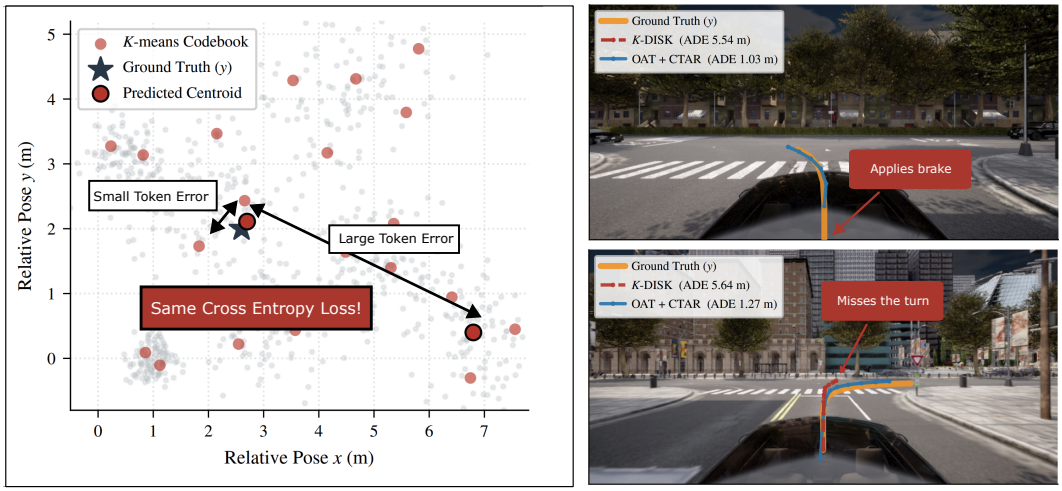


Figure 1: **Isometric cross-entropy loss geometry and closed-loop driving failure under unstructured tokenization (K-DISK). (Left)** Under standard categorical cross-entropy over a K-means codebook, predicting a nearby centroid (small physical error) and a distant centroid (large physical error) incurs the same loss penalty ($\mathcal{L}_{\text{CE}}$), creating a metric-agnostic, zero-gradient loss plateau off zero. **(Right)** Closed-loop failure modes on CARLA Bench2Drive: metric-agnostic cross-entropy causes K-DISK to suffer from false braking (*top*, 5.54 m ADE) and missed turns (*bottom*, 5.64 m ADE). In contrast, our proposed method (OAT + CTAR) maintains continuous trajectory tracking (1.03 m and 1.27 m ADE).

Under $\mathcal{L}_{\text{CE}}$, every incorrect token prediction is penalized identically. As illustrated in Figure 1 (left), this metric-agnostic supervision treats a sub-meter near-miss identically to a multi-meter prediction error. The uniform cross-entropy loss landscape fails to provide an effective signal to guide policy optimization toward physical ground truth, leading to closed-loop failure modes such as false braking and missed turns (Figure 1, right).

To resolve this optimization bottleneck without diverging from a token-native architecture, we introduce a framework that restores spatial metric geometry directly within the VLM token space via two complementary mechanisms:

1) We adapt Ordered Action Tokenization (OAT) (Liu et al., 2026a) from robotics domains to 2D trajectory planning, constructing an ordered, coarse-to-fine sequence topology where string prefix overlap correlates monotonically with continuous physical distance. Standard autoregressive sequence loss over OAT prefixes naturally imparts fine-grained metric feedback across sequence steps ($t > 1$).

2) To establish distance-awareness at the critical initial macro-decision boundary ($t = 1$) where cross-entropy remains flat, we introduce Coarse Trajectory Anchor Regularization (CTAR). CTAR decodes predicted soft token distributions through a frozen spatial decoder and backpropagates a smooth-L_1 trajectory distance loss directly into the VLM token embeddings using straight-through estimation.

Across extensive closed-loop and pseudo-closed-loop evaluations, our token-native policy resolves the discrete closed-loop failure, outperforming standard K-DISK by $+33.57$ DS on CARLA Bench2Drive and $+0.0979$ EPDMS on NAVSIM v2. Furthermore, OAT + CATR matches continuous actions heads (73.75 vs. 72.20 DS), proving that token-native VLAs can achieve state-of-the-art continuous control precision needed for closed-loop.

In summary, this paper makes three contributions:

1) We establish that discrete VLA performance breakdown is an optimization bottleneck caused by metric-agnostic cross-entropy ($\mathcal{L}_{\text{CE}}$), rather than an inherent limitation of tokenization.

2) We present a unified framework combining Ordered Action Tokenization (OAT) for prefix metric alignment with Coarse Trajectory Anchor Regularization (CTAR) to backpropagate differentiable spatial distance gradients into discrete action embeddings.

3) We demonstrate that embedding spatial metric feedback enables a token-native VLM to match the performance of specialized continuous control heads, demonstrating that VLAs can achieve closed-loop fidelity without departing from native token architectures.

2 Related Work

VLAs in Driving. End-to-end vision-language-action (VLA) architectures for autonomous driving have expanded across continuous, discrete, and hybrid paradigms (Shao et al., 2024; Sima et al., 2024; Renz et al., 2025; Zhou et al., 2026; Wang et al., 2026; Fu et al., 2026; Yang et al., 2026; Team & Inc, 2026; Feng et al., 2025). Continuous architectures append specialized regression heads (e.g., MLPs, GRUs, or query decoders) onto vision-language backbones to predict spatial waypoints (Shao et al., 2024; Sima et al., 2024; Renz et al., 2025; Fu et al., 2026; Liu et al., 2026b). While continuous heads preserve smooth trajectory outputs, they break the unified autoregressive token stream, preventing control actions from directly leveraging standard LLM serving optimizations such as KV-caching (Kwon et al., 2023), speculative decoding (Leviathan et al., 2023), and low-precision quantization. To reconcile language reasoning with control execution, recent hybrid frameworks like Alpamayo (Wang et al., 2026) utilize discrete action tokens for multi-step reasoning, but transition to continuous diffusion decoders for closed-loop execution. In contrast, we demonstrate that VLAs can achieve the precision needed for state-of-the-art closed-loop performance without departing from native Transformer architectures.

Action Tokenization. In contrast to previous approaches that adopt speacialized architectures for continuous action outputs, discrete token-native VLAs formulate trajectory planning as next-token prediction over quantized action spaces (Brohan et al., 2023; Zhou et al., 2026; Kim et al., 2024; Rowe et al., 2025; Liu et al., 2026a). Common discretization methods include per-dimension uniform binning (Brohan et al., 2023; Kim et al., 2024), K-means clustering (Zhou et al., 2026; Xiong et al., 2026), and vector-quantized codebooks (Wang et al., 2025). However, unstructured quantization constructs an isometric, metric-blind token space where categorical index distance bears no topological relationship to physical spatial displacement. Under standard cross-entropy, a minor trajectory error incurs the exact same penalty as a severe off-road deviation, yielding an uninformative zero-gradient plateau. To mitigate compounding execution errors under metric-blind supervision, prior works rely on temporal coarsening upsampled via splines (Rowe et al., 2025) or post-hoc reinforcement learning (Zhou et al., 2026). In contrast to these approaches, we adapt Ordered Action Tokenization (OAT) (Liu et al., 2026a) to 2D trajectory planning and introduce CTAR augmenting discrete token space with continuous spatial distance feedback across the entire action horizon.

3 Methodology

We formulate autonomous driving as a vision-conditioned sequence generation task. Given an input state context $\mathbf{s} \in \mathcal{S}$; comprising visual observations, ego-vehicle kinematics, and high-level navigational instructions; the VLA policy generates a future 2D trajectory $\mathbf{a} = \{(x_h, y_h)\}_{h=1}^{H} \in \mathbb{R}^{H\times 2}$ over a horizon H in the ego-centric reference frame. An overview of our complete framework, encompassing policy training with CTAR and the jointly-trained Ordered Action Tokenizer (OAT), is illustrated in Figure 2.

3.1 Metric-aware Tokenization for 2D Trajectory Planning

Standard token-native VLA architectures represent continuous trajectories via a sequence of language tokens (Rowe et al., 2025) or by fitting a K-means codebook $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ over relative trajectory poses (Zhou et al., 2026). A continuous trajectory $\mathbf{a}$ is quantized to a length-T sequence of its nearest centroids $\mathbf{c}^* = (c_1^*, \ldots, c_T^*)$. At each sequence step t, the VLM policy predicts

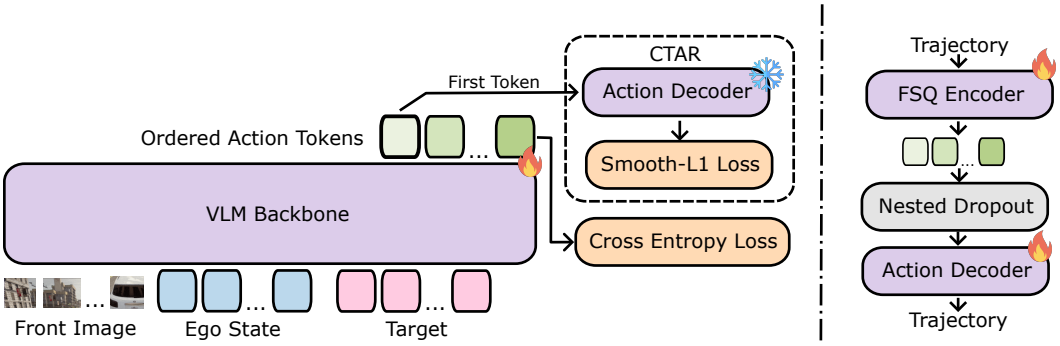


Figure 2: **(Left) Architecture and Policy Training:** The VLM backbone ingests front-camera frames, ego states, and target route conditioning to autoregressively predict an ordered action token sequence. The model is supervised by sequence Cross-Entropy Loss alongside our proposed CTAR, which extracts the initial coarse token and passes it through a frozen Action Decoder to compute a continuous Smooth-L_1 trajectory loss against trajectory ground truth. **(Right) Ordered Action Tokenizer (OAT):** The trajectory autoencoder employs a Transformer encoder-decoder with Finite Scalar Quantization (FSQ) and nested dropout across latent dimensions to enforce an ordered discrete codebook. Jointly trained under nested prefix masking, early tokens capture macro trajectory scale and orientation, while subsequent autoregressive tokens progressively refine spatial execution for flexible, coarse-to-fine trajectory detokenization.

a logit distribution $\mathbf{z}^{(t)} \in \mathbb{R}^K$ over vocabulary $\mathcal{C}$, optimized using sequence cross-entropy:

$$\mathcal{L}_{\text{CE}}(\mathbf{z}, \mathbf{c}^*) = \sum_{t=1}^{T} -\log P(c_t^* \mid c_{<t}^*) = \sum_{t=1}^{T} \ell_t = \sum_{t=1}^{T} -\log \left(\frac{e^{z_{c_t^*}^{(t)}}}{\sum_{j=1}^{K} e^{z_j^{(t)}}} \right). \tag{1}$$

Evaluating the gradient of $\mathcal{L}_{\text{CE}}$ with respect to any non-target logit $z_k^{(t)}$ $(k \neq c_t^*)$ yields:

$$\frac{\partial \mathcal{L}_{\text{CE}}}{\partial z_k^{(t)}} = \frac{\partial \ell_t}{\partial z_k^{(t)}} = P(k \mid c_{<t}^*). \tag{2}$$

The gradient magnitude $\frac{\partial \mathcal{L}_{\text{CE}}}{\partial z_k^{(t)}}$ at step t depends exclusively on the predicted conditional probability $P(k \mid c_{<t}^*)$ and carries zero information regarding the physical metric distance $\|\mathbf{c}_k - \mathbf{c}_{c_t^*}\|_2$ between candidate centroid $\mathbf{c}_k$ and target centroid $\mathbf{c}_{c_t^*}$. Hence, under an unstructured codebook, categorical cross-entropy induces an *isometric loss geometry*: for any distinct target-prediction pair $(i \neq j)$, the penalty distance is binary $(d_{\text{loss}}(c_i, c_j) = 1)$. Under this geometry, a small trajectory error and a large error generate identical loss penalties, depriving the policy of the continuous "near-miss" gradient feedback required for closed-loop control and recovery.

To represent continuous trajectories $\mathbf{a}$ within an autoregressive LLM vocabulary without destroying spatial distance relationships, we adapt Ordered Action Tokenization (OAT) (Liu et al., 2026a) to continuous 2D driving trajectories. OAT maps continuous waypoints $\mathbf{a}$ into an ordered sequence of T discrete tokens, $\mathbf{z} = (z_1, z_2, \ldots, z_T) \in \mathcal{V}^T$ via an encoder, and reconstructs them using a decoder D_ϕ, where $\mathcal{V}$ denotes the action vocabulary.

To induce a coarse-to-fine hierarchy, the tokenizer is trained using nested prefix masking (Figure 2, right), minimizing the reconstruction error of partial sequence prefixes $z_{1:k}$ against ground-truth waypoints $\mathbf{a}$:

$$\mathcal{L}_{\text{codec}} = \sum_{k=1}^{T} \|\mathbf{a} - D_\phi(z_{1:k}, \mathbf{0}_{>k})\|_2^2 . \tag{3}$$

Because the expected prefix reconstruction error $\epsilon_k = \mathbb{E}\left[\|\mathbf{a} - D_\phi(z_{1:k}, \mathbf{0}_{>k})\|_2\right]$ decreases strictly with prefix length k, OAT induces a prefix topology, such that any two trajectories $\mathbf{a}_A$ and $\mathbf{a}_B$ sharing a k-length prefix map to the same coarse anchor $D_\phi(z_{1:k})$, bounding their physical distance

by $\|\mathbf{a}_A - \mathbf{a}_B\|_2 \leq 2\epsilon_k$. Consequently, string-prefix divergence serves as a proxy for physical spatial distance, i.e. trajectories with shorter common prefixes (larger d_{string}) correspond to a higher upper bound on their maximum physical separation $\|\mathbf{a}_A - \mathbf{a}_B\|_2$.

When paired with sequence-level cross entropy loss $\mathcal{L}_{\text{CE}}(\mathbf{z}^*)$, this prefix topology transforms categorical cross-entropy into a distance-aware loss for steps $t > 1$:

- **Near-Miss Predictions ($\mathbf{a}_{\text{near}}$):** Shares an initial prefix of length k with target $\mathbf{a}^*$ ($z_1 = z_1^*, \ldots, z_k = z_k^*$). For $t \leq k$, the model predicts correct prefix tokens, yielding $\ell_t \approx 0$. Loss accumulates only over the remaining $T - k$ refinement steps: $\mathcal{L}_{\text{CE}}(\mathbf{a}_{\text{near}}) \approx \sum_{t=k+1}^{T} \ell_t$, scaling roughly with the number of diverging steps ($T - k$).
- **Far-Miss Predictions ($\mathbf{a}_{\text{far}}$):** Diverges at the first coarse token ($z_1 \neq z_1^*$). Conditioning on an incorrect prefix shifts the autoregressive context, accumulating high cross-entropy loss across all T steps: $\mathcal{L}_{\text{CE}}(\mathbf{a}_{\text{far}}) = \sum_{t=1}^{T} \ell_t$, spanning the full horizon T.

Because the effective horizon of accumulated error scales with string-prefix divergence ($T - k$), sequence-level cross-entropy over OAT natively provides distance-graded feedback:

$$\mathcal{L}_{\text{CE}}(\mathbf{a}_{\text{near}}) \ll \mathcal{L}_{\text{CE}}(\mathbf{a}_{\text{far}}). \tag{4}$$

3.2 Coarse Trajectory Anchor Regularization (CTAR)

While OAT provides metric feedback across sequence steps $t > 1$, an optimization bottleneck persists at step $t = 1$. Because the initial coarse token z_1 has no preceding prefix ($z_{<1} = \emptyset$), cross-entropy loss at step 1 ($\ell_1 = -\log P(z_1^*)$) remains strictly isometric across top-level macro categories. If the policy mispredicts the first coarse token, pure cross-entropy treats all incorrect top-level coarse categories as uniformly wrong. To anchor the policy's initial macro decision boundary directly in physical metric geometry, we introduce Coarse Trajectory Anchor Regularization (CTAR; Figure 2, left).

At sequence position $t = 1$ predicting the initial coarse token z_1, the policy's logit distribution $\mathbf{z}^{(1)} \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary is relaxed into a soft probability distribution $p = \text{softmax}(\mathbf{z}^{(1)})$. We construct a differentiable one-hot code representation $\mathbf{v}_{\text{st}}$ using a Straight-Through Estimator (STE) (Bengio et al., 2013):

$$\mathbf{v}_{\text{st}} = p + \Big(\mathbf{e}_{\arg\max p} - p\Big)_{\text{detach}}, \tag{5}$$

where $\mathbf{e}_k$ is a standard basis vector. The straight-through code $\mathbf{v}_{\text{st}}$ is mapped to FSQ codebook latent embeddings $E \in \mathbb{R}^{|\mathcal{V}| \times D}$:

$$\mathbf{z}_{\text{st}} = \mathbf{z}_{\text{hard}} + \Big(\mathbf{v}_{\text{st}} E - (\mathbf{v}_{\text{st}} E)_{\text{detach}}\Big). \tag{6}$$

The relaxed coarse latent $\mathbf{z}_{\text{st}}$ (padded with zeros for $t > 1$) is passed through the tokenizer's decoder D_ϕ to yield decoded physical waypoints $\hat{\mathbf{a}} = D_\phi(\mathbf{z}_{\text{st}}, \mathbf{0}_{>1}) \in \mathbb{R}^{H \times 2}$. We compute a smooth-$L_1$ metric loss against ground-truth trajectory $\mathbf{a}^*$:

$$\mathcal{L}_{\text{CTAR}} = \frac{1}{2H} \sum_{h=1}^{H} \sum_{d=1}^{2} \text{Smooth}_{L_1}\Big(\hat{a}_{h,d} - a^*_{h,d};\, \beta\Big), \tag{7}$$

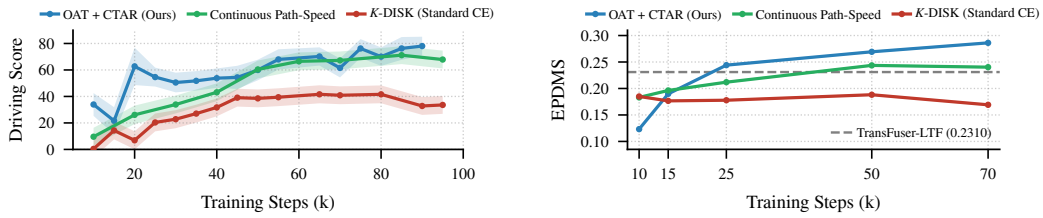
where $\beta = 0.02\,\text{m}$ defines a tight quadratic-to-linear error threshold.

The complete policy optimization objective combines language cross-entropy $\mathcal{L}_{\text{CE}}^{\text{lang}}$, trajectory token cross-entropy $\mathcal{L}_{\text{CE}}^{\text{traj}}$, and the metric anchor loss:

$$\mathcal{L}_{\text{total}}(t) = \mathcal{L}_{\text{CE}}^{\text{lang}} + \mathcal{L}_{\text{CE}}^{\text{traj}} + \lambda_{\text{CTAR}}(t) \cdot \mathcal{L}_{\text{CTAR}}, \tag{8}$$

where $\lambda_{\text{CTAR}}(t) = \text{CosineBlend}(0.0 \to 0.1,\, t/T_{\text{max}})$ ramps the regularization weight smoothly from 0.0 to 0.1 over training duration T_{max}.

During policy optimization under $\mathcal{L}_{\text{total}}$, the tokenizer decoder weights ϕ are kept frozen, such that $\nabla_\phi \mathcal{L}_{\text{CTAR}} = \mathbf{0}$. Freezing D_ϕ prevents codebook collapse, forcing the VLM backbone to align its internal hidden representations with physical metric geometry.



(a) CARLA Bench2Drive-44 (b) NAVSIM v2

Figure 3: **Performance progression across training iterations** (shaded: 95% CI). **(a)** CARLA Driving Score: OAT + CTAR consistently leads K-DISK and continuous baselines. **(b)** NAVSIM v2 EPDMS: While baselines plateau early, OAT + CTAR exhibits sustained progression, reaching 0.2861 EPDMS at 70k steps.

4 RESULTS

We evaluate our framework across two core questions:

a) Does restoring spatial metric feedback via OAT + CTAR resolve the closed-loop failure of token-native VLAs under cross-entropy?

b) Can a token-native VLA match or outperform dedicated continuous regression heads under identical backbones and training protocols?

4.1 EXPERIMENTAL SETUP

Evaluation Benchmarks. We evaluate across two complementary autonomous driving benchmarks. CARLA Bench2Drive (Jia et al., 2024) is a closed-loop simulator where the primary metric, Driving Score (DS), penalizes route completion by infractions (e.g., collisions, lane departures). We report results on the full 220-route suite as well as a 44-route stratified subset (`scenario44`, containing one route per scenario category) used for checkpoint-level analysis. NAVSIM v2 (Dauner et al., 2024) evaluates policies in a pseudo-closed-loop environment across 5,912 validation scenes from the `navhard_two_stage` split using the Extended Predictive Driver Model Score (EPDMS), providing a trajectory simulation metric that tracks closed-loop driving fidelity significantly better than open-loop displacement error.

Baselines and Action Interfaces. All policies share an identical `InternVL2-1B` backbone (Chen et al., 2024), receiving a single front-camera frame (1,024 visual tokens) interleaved with a text prompt specifying current ego-velocity and navigational goals. Target action geometries are benchmark-dependent: CARLA predicts 10 waypoints at 0.2 s intervals (2.0 s horizon) in continuous 2D space (x, y), whereas NAVSIM v2 predicts 8 waypoints at 0.5 s intervals (4.0 s horizon) in 3D space (x, y, θ). Evaluated interfaces differ strictly in their action output representations:

- **K-DISK (Zhou et al., 2026):** The standard discrete VLA interface, emitting 10 (CARLA) or 8 (NAVSIM) tokens indexing a frozen codebook of 2,048 centroids fitted over relative poses, supervised strictly via sequence categorical cross-entropy ($\mathcal{L}_{\text{CE}}$).
- **Continuous Path-Speed Head (Renz et al., 2025):** Continuous control baseline using MLP regression heads to predict spatial waypoints (xy) and longitudinal speeds directly (heading derived via finite-difference tangents). *Benchmark Adaptation:* On NAVSIM v2 the route-prediction head is disabled (the converted corpus derives routes from the ego's own future poses, so it would regress a copy of its trajectory label); the 8-pose spatial path and speeds are both emitted by the single cumulative-regression speed head, with target-point and command text retained as prompt conditioning.
- **OAT + CTAR (Ours):** Our proposed token-native interface emitting 32 ordered coarse-to-fine discrete tokens from an FSQ codebook ($[8, 5, 5, 5]$ grid layout, 1,000 valid codes per level) paired with a causal register decoder, supervised by $\mathcal{L}_{\text{CE}}$ and the $\mathcal{L}_{\text{CTAR}}$ metric anchor. Crucially, the tokenizer codec architecture is identical across both benchmarks.

Table 1: **Closed-loop results on the Bench2Drive-220 suite** (1,600 s wall-clock budget, identical agent wrapper). DS is the clean mean Driving Score; RC is Route Completion (%); Pen. is the infraction penalty factor; Compl. is full-route completion rate (%). Infraction columns report raw counts across observed routes.

	Primary Driving Metrics				Infraction Counts ↓				
Policy	**DS ↑**	**RC ↑**	**Pen. ↑**	**Compl. ↑**	**Veh.**	**Ped.**	**Red**	**Stop**	**OOB**
OAT + CATR (ours)	**73.75**	**82.2**	**0.846**	**66%**	**55**	2	6	3	**14**
SimLingo (official)	72.20	82.1	0.844	59%	65	**0**	5	4	21
Continuous baseline	63.44	73.3	0.819	47%	63	2	**4**	**1**	37
K-DISK	37.98	52.2	0.718	24%	81	6	21	7	41

Datasets and Training Corpora. Policies are trained on multimodal driving datasets across both environments. For CARLA Bench2Drive, the corpus comprises 37 primary trajectory chunks and 25 world-model dreamer chunks (Renz et al., 2025), spanning 7 scenario-specific `lb1_split` route families (*e.g.*, `DynamicObjectCrossing`, `OppositeVehicleRunningRedLight`, `SignalizedJunction`), a 150-route single-scenario set, a 100-route three-scenario set, and a Town12 parking-lane set. Across ≈ 825 unique routes averaging ≈ 95 frames per route, each training epoch processes $\approx 66{,}000$ frames (8,257 steps at batch size 8), totaling $\approx 991{,}000$ processed samples over 15 epochs. For NAVSIM v2, models are trained on 14 chunks of the `navtrain` trainval split ($\approx$ 59,600 scenes). Training executes over 70k optimization steps (8 samples per step), processing $\approx 40{,}000$ samples per optimization epoch (5,000 steps) and totaling $\approx 560{,}000$ processed samples across 9.4 dataset passes. Additional details are provided in Appendix A.1

Training, Hardware, and Implementation Details. All models fine-tune an `InternVL2-1B` backbone (Chen et al., 2024) with LoRA adapters (Hu et al., 2021) across 8×NVIDIA A100 (80 GB) GPUs using DeepSpeed ZeRO-2 (Rasley et al., 2020). Training executes over 15 epochs using a OneCycle learning rate schedule (Smith, 2018), a per-device batch size of 8, and a fixed seed (9876). Arms are strictly matched within each benchmark across corpus, seed, batch size, and optimization schedule, with two noted disclosures: evaluation checkpoint cadence is 5k steps on CARLA versus 2.5k steps on NAVSIM v2, and the NAVSIM K-means arm was trained in fp16 precision compared to bfloat16 for its siblings (a minor asymmetry that handicaps the K-means baseline, rendering its reported underperformance conservative). Closed-loop CARLA evaluations execute behind a byte-identical agent wrapper with an extended 1,600 s per-route wall-clock budget to eliminate asymmetric censoring artifacts.

4.2 Closed-Loop Performance on CARLA Bench2Drive

We evaluate closed-loop driving performance on CARLA Bench2Drive to address our primary empirical claims regarding discrete K-means tokenization and continuous control heads.

Comparison Against K-DISK. On the 44-route evaluation suite (`scenario44`) at matched training progress (105k steps), OAT + CATR achieves 75.19 Driving Score (DS) compared to 35.50 DS for K-DISK (+39.69 DS gain). As illustrated in the optimization trajectory (Figure 3a), comparing each policy at its peak checkpoint yields 83.5 DS for OAT + CATR versus 45.8 DS for K-DISK (+37.7 DS gain). Across all experiments, K-DISK is evaluated under time-aware control decoding, correcting a legacy start-from-rest controller defect that artificially depresses scores by applying braking forces even to ground-truth trajectories (Appendix A.8).

Comparison Against Continuous Baselines. Table 1 presents evaluation results across the full 220-route Bench2Drive suite under a 1,600 s wall-clock limit. OAT + CATR reaches 73.75 DS, outperforming our continuous path-speed baseline (63.44 DS) and achieving statistical parity with the official released continuous SimLingo checkpoint (72.20 DS). On common routes, the paired difference between OAT + CATR and the official checkpoint is $+2.14 \pm 5.20$ DS ($n = 211$), falling strictly within the paired detection threshold.

Table 2: **Open-loop results on NAVSIM v2** (`navhard_two_stage`, full 5,912-scene split). EPDMS is the two-stage composite score; remaining columns report component scores: **NC** (No At-fault Collision), **DAC** (Drivable Area Compliance), **DDC** (Driving Direction Compliance), **EP** (Ego Progress), **TTC** (Time-To-Collision), **LC** (Lane Center Compliance), **C** (Comfort), and **EC** (Extended Comfort).

	Composite	**Stage-One Component Sub-Metrics ↑**							
Policy	**EPDMS ↑**	**NC**	**DAC**	**DDC**	**EP**	**TTC**	**LC**	**C**	**EC**
OAT + CATR (ours)	**0.2861**	0.951	**0.853**	**0.984**	**0.840**	**0.942**	**0.964**	0.962	0.373
Continuous baseline	0.2403	**0.959**	0.753	0.992	0.840	0.931	0.940	**0.976**	**0.747**
K-DISK	0.1692	0.942	0.753	0.946	0.824	0.922	0.907	0.913	0.516

Calibration Against Official Continuous Weights. When evaluated on our execution harness, the official released SimLingo checkpoint achieves 72.20 DS compared to its published value of 85.07 DS. This delta reflects evaluation infrastructure differences rather than model implementation: the official checkpoint scores 72.60 DS on our 44-route subset and 72.10 DS on the 173 remaining routes, demonstrating uniform performance across the suite. We therefore establish 72.20 DS as the calibrated continuous reference for all like-for-like comparisons.

4.3 Pseudo-Closed-Loop Performance on NAVSIM v2

We evaluate all three action interface paradigms on NAVSIM v2 across training iterations (Figure 3b and Table 2).

At 10k training steps, K-DISK (0.1849 EPDMS) and the continuous path-speed head (0.1831 EPDMS) are statistically tied within the measured 0.002121 noise floor. This confirms that discrete action tokenization itself imposes no inherent performance penalty when initial codec reconstruction fidelity is high. Conversely, OAT + CATR begins at 0.1232 EPDMS, a lower initial score directly dictated by its coarse early reconstruction ceiling (1.182 m ADE at step 10k).

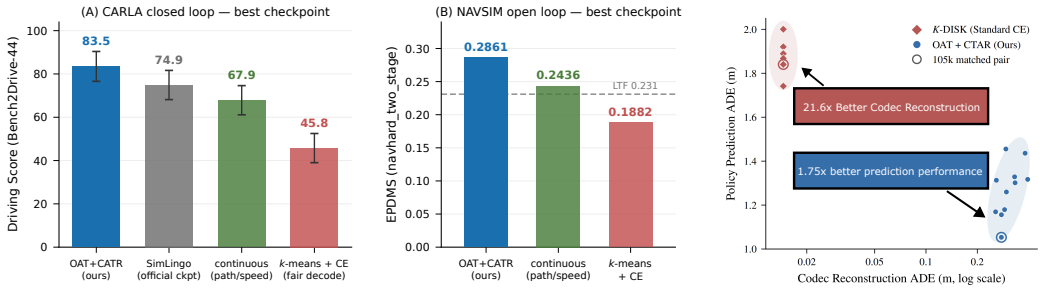
Between 15k and 25k steps, a complete performance inversion occurs across the three interfaces (Figure 3b). As the VLM backbone learns to leverage the ordered prefix topology, OAT + CATR advances rapidly, surpassing the published TransFuser-LTF reference baseline (0.2310 EPDMS (**?**)) at 25k steps (0.2441 EPDMS) and reaching 0.2861 EPDMS at 70k steps.

At the matched 70k checkpoint, OAT + CATR achieves a $+0.0458$ EPDMS gain over the continuous path-speed head ($21.6\times$ the noise floor) and a $+0.1169$ EPDMS lead over K-DISK ($55.1\times$ the noise floor). Comparing each interface at its respective peak checkpoint yields a $+0.0425$ EPDMS lead over continuous control (0.2861 vs. 0.2436) and a $+0.0979$ EPDMS gain over K-DISK (0.2861 vs. 0.1882).

4.4 Representation Ceiling vs. Policy Prediction

To verify that K-DISK underperformance stems from isometric optimization rather than representational capacity, we evaluate 288 real CARLA frames across training checkpoints (10k–115k steps), decoupling model-free reconstruction error (**Codec ADE**) from policy emission error (**Predicted ADE**).

As shown in Figure 4b, K-DISK provides $21.6\times$ higher reconstruction fidelity than OAT (0.0146 m vs. 0.315 m ADE). Yet, K-DISK consistently yields higher policy prediction error (1.840 m vs. 1.053 m ADE at 105k steps), overshooting its representational ceiling by $126\times$ (compared to just $3.3\times$ for OAT + CTAR). This policy deficit concentrates in longitudinal velocity estimation. Specifically, in pull-away scenarios, K-DISK exhibits false braking in 50.0% of cases (vs. 22.7% ground truth), whereas OAT + CTAR aligns closely with ground truth (27.3%). Because small velocity changes map to arbitrary token index shifts in unstructured codebooks, categorical cross-entropy provides no distance-aware corrective gradients for near-miss speed predictions—a failure directly repaired by CTAR's continuous metric anchor.



(a) Peak benchmark performance (b) Representation vs. policy error

Figure 4: **Empirical evaluation summary across CARLA Bench2Drive and NAVSIM v2.** **(a)** Peak Driving Score on CARLA `scenario44` (left) and EPDMS on NAVSIM `navhard_two_stage` (right). OAT + CTAR substantially outperforms K-DISK (+37.7 DS, +0.0979 EPDMS) and matches/exceeds continuous control heads. **(b)** Representation ADE vs. policy prediction error (10k–115k steps; rings mark 105k pair). Despite a 21.6× coarser codec, OAT + CTAR yields 1.75× lower prediction error, overshooting its representation ceiling by only 3.3× (vs. 126× for K-DISK).

4.5 Ablations

Impact of CTAR. As shown in Table 3, removing the CTAR metric anchor causes significant performance degradation, trailing both full OAT + CATR and continuous baselines. Ordered token sequences require explicit, differentiable distance feedback to prevent cross-entropy optimization plateaus.

Impact of Frozen-Decoder Guardrail. Unfreezing the tokenizer decoder (D_ϕ) triggers severe codebook collapse, drastically reducing unique active trajectories and increasing lateral displacement error (see Table 4). Keeping D_ϕ strictly frozen forces the VLM backbone to align internal states with spatial metric geometry.

Codec Capacity Scaling. Increasing FSQ codebook capacity yields virtually identical reconstruction error while codebook utilization plunges (see Table 5). Residual reconstruction error represents an architectural ceiling rather than a capacity bottleneck.

Table 3: CTAR Loss Impact

Configuration	**DS ↑**
OAT + CTAR	**75.19**
Continuous baseline	63.44
OAT w/o CTAR	48.50

Table 4: D_ϕ Guardrail

D_ϕ **State**	**Seq. ↑**	**Lat. Err. ↓**
Frozen	**675**	**1.0×**
Unfrozen	52	6.2×

Table 5: FSQ Capacity

Codes	**ADE (m) ↓**	**Util. ↑**
1,000	**0.384**	**82%**
17,280	0.396	13%

5 Conclusion

In this work, we establish that closed-loop execution failures in token-native VLAs stem from an optimization bottleneck induced by the isometric nature of cross-entropy loss, rather than a fundamental limitation of tokenization. Standard action tokenization (K-DISK) constructs a metric-agnostic loss landscape where near-misses and large deviations incur identical penalties, causing policy failure on CARLA Bench2Drive (41.62 Driving Score). To resolve this optimization mismatch while preserving a token-native Transformer architecture, we introduce a unified framework pairing OAT for prefix metric alignment with a proposed CTAR loss, which backpropagates continuous spatial distance gradients into token embeddings. Across extensive evaluations on CARLA Bench2Drive and NAVSIM v2, our OAT + CTAR reaches 75.19 DS and 0.2861 EPDMS (+33.57 and +0.0979 over K-DISK, respectively), as well as matches the performance of specialized continuous regression heads while maintaining complete compatibility with standard LLM serving infrastructure. These results demonstrate that token-native action streams can deliver control precision comparable to continuous models, offering a principled, scalable path forward for embodied foundation models.

AI USE STATEMENT

In this work, we used generative AI tools for the initial drafting of the manuscript and assistance with writing code. We have not used generative AI tools for formulating core research hypotheses or autonomous scientific discovery, and those other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for assistance in coding and debugging during the implementation of experiments. We have reviewed all AI-assisted work. Specifically, all AI-generated code and experiments were conducted under active human supervision, thoroughly tested for correctness, and carefully verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility and ensure transparency, we have documented our experimental setup, model architectures, and training procedures across the main paper and appendix. Specifically, detailed descriptions of the dataset collection, processing steps, and core training methodology are provided in Section 4 of the main text. Comprehensive implementation details, complete hyperparameter configurations, and further architectural specifications are fully outlined in the Appendix. Furthermore, the complete training and evaluation source code, alongside pre-trained model checkpoints, will be released publicly in the near future upon securing the necessary institutional and legal approvals.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, June 2024.

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.

Bowen Feng, Zhiting Mei, Julian Ost, Filippo Ghilotti, Baiang Li, Roger Girgis, Anirudha Majumdar, and Felix Heide. Verdi: Vlm-embedded reasoning for autonomous driving. *arXiv preprint arXiv:2505.15925*, 2025.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Dingkang Liang, and Xiang Bai. Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. In *European Conference on Computer Vision*, pp. 503–522. Springer, 2026.

Google DeepMind. Gemini Robotics: Embodied Reasoning. `https://deepmind.google/models/gemini-robotics/embodied-reasoning/`, 2025. Accessed: 2026-09-23.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. Oat: Ordered action tokenization. *arXiv preprint arXiv:2602.04215*, 2026a.

Qi Liu, Yabei Li, Hongsong Wang, Heng Zhang, and Lei He. Anchorvla: Bridging discrete decisions and continuous trajectories for vision-language-action planning. *arXiv preprint arXiv:2607.03182*, 2026b.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

OpenAI. GPT-6 Astra. `https://openai.com/index/gpt-6-astra/`, 2026. Accessed: 2026-09-23.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

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

Luke Rowe, Rodrigue de Schaetzen, Roger Girgis, Christopher Pal, and Liam Paull. Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. *arXiv preprint arXiv:2506.11234*, 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130. IEEE, 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 *European conference on computer vision*, pp. 256–274. Springer, 2024.

Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

Foundation Model Team and XPeng Inc. Xcot-vla: Executable chain-of-thought for vision-language-action driving. *arXiv preprint arXiv:2608.10976*, 2026.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail, 2026. URL `https://arxiv.org/abs/2511.00088`.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11089–11099. IEEE, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Zhen Xiong, Shang-Ling Hsu, and Cyrus Shahabi. Trajtok: Adaptive spatial tokenization for trajectory representation learning. *arXiv preprint arXiv:2605.20134*, 2026.

Nan Yang, Zhanwen Liu, Linfeng Zhang, Shangyu Xie, Yang Wang, Wenzhuo Zhou, and Xiangmo Zhao. Mvpruner: Dynamic token pruning for accelerating multi-view vision-language models in autonomous driving. *arXiv preprint arXiv:2606.27660*, 2026.

Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. In *European Conference on Computer Vision*, pp. 336–353. Springer, 2026.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026.

A Appendix

A.1 Shared Policy Architecture and Training

All policy variants share the same foundational architecture. A front camera frame is encoded into 1,024 visual tokens and spliced into the language model's input embeddings. The input prompt provides the ego vehicle's current speed and target waypoint, concluding with a fixed assistant turn token `Waypoints:`.

All models are trained using bf16 mixed precision via DeepSpeed ZeRO-2 (Rasley et al., 2020) and a OneCycle learning rate schedule. For NAVSIM, the training corpus consists of 152,495 unmixed anchors (step 10,000 corresponds to epoch 2.10); for CARLA, all arms are trained on identical corpus lists under identical parity modes with commentary and QA enabled. Configurations for trajectory and continuous arms are identical except for the output head and routing flags.

A.2 Action Interfaces

- **k-means action tokens (`vlm_kdisk`):** A frozen k-means codebook over relative poses $(\Delta x, \Delta y, \Delta\psi)$ in footprint-scaled coordinates encodes trajectories via greedy nearest-neighbor rollout; decoding is the exact inverse. NAVSIM uses $K = 2048$ with 8 tokens per 4 s (88 bits total). The policy is optimized using standard categorical cross-entropy ($\mathcal{L}_{\text{CE}}$).
- **Ordered Trajectory Tokens (OAT):** Comprises an encoder-quantizer-decoder codec featuring a depth-2 Transformer encoder, a Finite Scalar Quantization (FSQ) bottleneck with levels $[8, 5, 5, 5]$ (1,000 codes), and a depth-3 causal register decoder with 32 registers at width 256. The 32 codes are ordered coarse-to-fine, yielding an effective capacity of ≈ 319 bits.
- **Continuous Regression (`path_speed`):** Employs adaptor queries feeding dedicated path and speed heads without an intermediate action vocabulary, utilizing finite-difference tangent heading reconstruction at evaluation.

A.3 The Prefix Property and Metric Placeability

The topological structure of OAT enables explicit metric supervision. Evaluating 92,235 pairs of real NAVSIM anchors, edit-distance similarity on the 32-code sequence correlates strongly with physical Euclidean distance ($\rho = -0.670$), compared to $\rho = -0.170$ for k-means code identity (Figure 5).

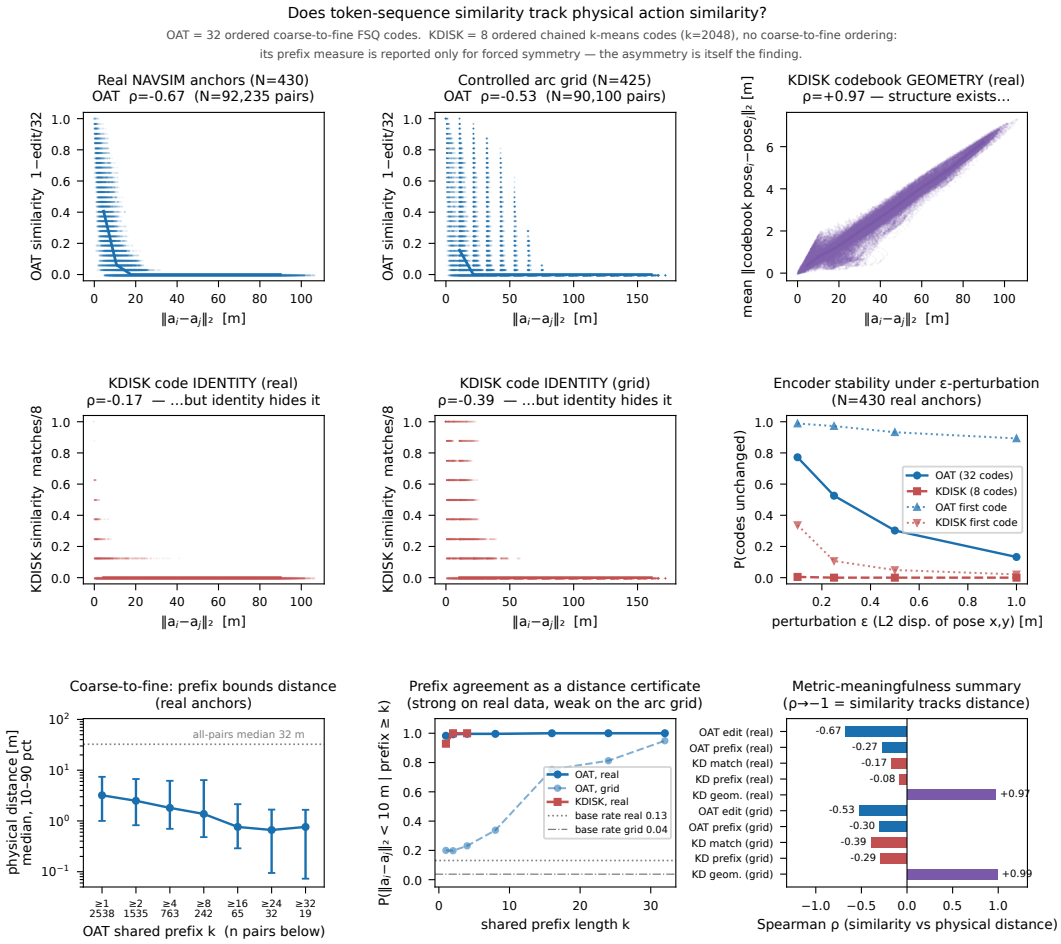


Figure 5: **Token-sequence similarity vs. physical spatial distance. (Top/Middle)** Comparison of Ordered Action Tokenizer (OAT, 32 codes) against k-means (K-DISK, 8 codes). OAT token sequence edit distance strongly tracks physical trajectory Euclidean distance ($\rho = -0.670$ on real NAVSIM anchors), whereas k-means code identity fails to capture spatial proximity ($\rho = -0.170$). OAT first codes remain highly stable under spatial ϵ-perturbations. **(Bottom)** Shared OAT prefixes act as a deterministic spatial distance certificate ($P(< 10\,\text{m} \mid \text{prefix} \geq 1) = 0.983$), where increasing prefix depth k monotonically bounds physical trajectory deviation.

Specifically, when two trajectories share an initial code prefix ($k \geq 1$), their physical distance exhibits a median of 3.2 m and a 95th percentile of 8.6 m (maximum 12.5 m across 2,538 pairs), yielding a spatial distance certificate of $P(\|\mathbf{a}_i - \mathbf{a}_j\|_2 < 10\,\text{m} \mid \text{prefix} \geq 1) = 0.983$ (AUC 0.977) against a base rate of 0.131. Raw prefix correlation is weaker ($\rho = -0.270$) due to zero-inflation from rare prefix agreement (absent in 97.3% of random real pairs). On a synthetic constant-curvature arc grid, the first-code certificate drops initially ($P = 0.200$) and recovers monotonically with depth.

A.4 Benchmarks and Evaluation Protocol

`navhard_two_stage` / EPDMS (NAVSIM v2). Evaluated on the two-stage pseudo-closed-loop split (76 logs, 5,912 tokens) with reactive traffic. Extended Predictive Driver Model Score (EPDMS) is a composite multiplicative metric: EPDMS $=$ NC$\cdot$DAC$\cdot$TLC$\cdot$DDC$\cdot(5\,\text{EP}+5\,\text{TTC}+2\,\text{LC}+2\,\text{C}+2\,\text{EC})/18$. All reported numbers evaluate the full split (5,912/5,912 valid scenarios, 0 failures).

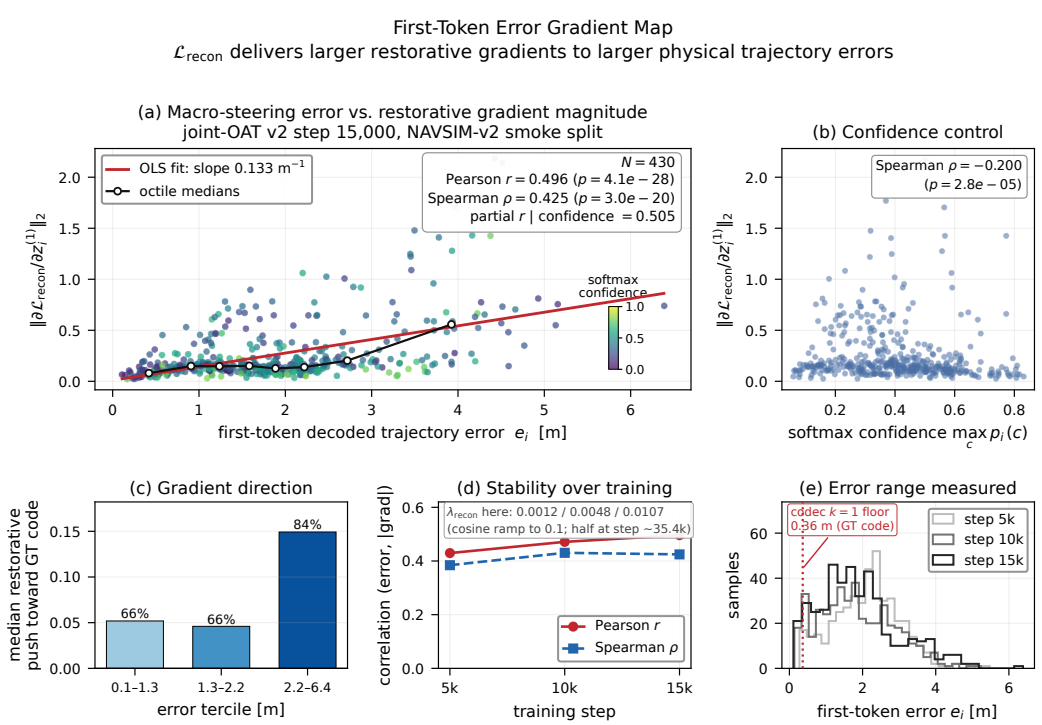


Figure 6: **CTAR gradient mapping and restorative properties.** **(a)** Restorative gradient magnitude $\|\partial \mathcal{L}_{\text{recon}}/\partial \mathbf{z}_i^{(1)}\|_2$ scales linearly with first-token decoded trajectory error e_i (Pearson $r = 0.496$, OLS slope $0.133\,\text{m}^{-1}$). **(b)** Controlling for softmax confidence confirms the gradient magnitude is explicitly driven by physical trajectory error (partial $r = 0.505$). **(c)** Gradient direction provides the largest restorative push (84%) during large macro-steering errors ($> 2.2\,\text{m}$). **(d–e)** Error-gradient correlations remain stable across training steps, operating above the discrete codec error floor ($0.36\,\text{m}$).

CARLA / Bench2Drive Driving Score. Evaluated on Bench2Drive route suites (Jia et al., 2024): `scenario44` (stratified one-route-per-scenario subset), and `b2d220`. All suites share the underlying `bench2drive220.xml` specification, rendering comparisons against published baselines direct. All CARLA metrics are recomputed from 4,073 sanitized per-route records.

A.5 Empirical Noise Floors and Non-Determinism

NAVSIM Floor (0.002121 **EPDMS).** Evaluated by comparing KV-cached versus uncached greedy decoding for an identical checkpoint. Uncached re-scoring exhibits zero run-to-run variance (12/12 bitwise identical), establishing 0.002121 EPDMS (0.047393 vs. 0.049514) as the strict upper bound on numerical decode-path artifacts.

CARLA Floor ($\sigma_{\text{route}} = 22.81$ **DS).** Evaluated across 62 verified, bitwise-identical agent code replicates under identical scene setups. The route score deviation distribution is bimodal: 60% of routes are bitwise identical, 18% exhibit catastrophic score flips ($|\Delta\text{DS}| \geq 60$), and 27% experience completion status flips (Completed $\leftrightarrow$ Failed). This yields a pooled per-route standard deviation of $\sigma_{\text{route}} = 22.81$ DS. The resulting 95% minimum detectable gaps (MDG) are **29.8 DS** ($n = 5$), **10.8 DS** ($n = 44$), and **6.2 DS** ($n = 220$).

Root Cause of CARLA Variance. While simulator parameters are fully deterministic (synchronous mode, fixed Δt, seeded traffic manager), standard agent evaluation scripts set `cudnn.benchmark=True`, `cudnn.deterministic=False`, and omit PyTorch RNG seed-

ing. Under multi-GPU contention, non-deterministic kernel execution introduces sub-millimeter waypoint shifts that compound over closed-loop rollouts, triggering discrete collision boundary flips.

A.6 NAVSIM Evaluation Issues

Open-loop and pseudo-closed-loop benchmarks require strict consistency between training and evaluation pipelines. We identify two critical evaluation protocol hazards that systematically distort baseline comparisons, particularly between discrete and continuous action interfaces:

KV-Cache Sequence Contamination. In downstream VLA evaluation harnesses, unconstrained language sampling executed prior to action decoding can emit unconditioned pseudo-language tokens directly into the model's key-value (KV) cache. For discrete token-native policies, subsequent action codes are forced to condition on this unconditioned prefix, severely degrading downstream trajectory prediction. Continuous action heads—which terminate immediately upon emitting an end-of-sequence token—are naturally immune to this prefix noise, creating an artificial performance gap between continuous and discrete architectures. Ensuring exact prompt-tail alignment between training and inference eliminates this evaluation bias.

Visual Preprocessing Discrepancy. Omission of training-time visual preprocessing during evaluation—such as failing to apply vertical cropping ($1080 \rightarrow 756$)—introduces a significant tensor distribution shift between training and test inputs. Re-aligning visual transformation pipelines guarantees strictly zero domain shift across all action modalities.

As detailed in Table 6, resolving both evaluation hazards simultaneously yields a 62% improvement beyond the sum of isolated repairs. This interaction demonstrates that unaddressed evaluation defects non-linearly compound, disproportionately suppressing discrete VLA performance and obscuring true model capabilities.

Table 6: **NAVSIM evaluation** (`navhard_two_stage` full split, step 10k). Aligning inference prompts and input image transformations restores true policy performance.

Action Interface	Pre-fix EPDMS	Post-fix EPDMS	Relative Gain	Stage-1 Score (Pre → Post)
`vlm_kdisk`	0.0495	**0.1849**	$3.73\times$	$0.2324 \rightarrow 0.4865$
`path_speed`	0.1188	**0.1831**	$1.54\times$	$0.3644 \rightarrow 0.4907$
`joint_oat`	0.1019	**0.1232**	$1.21\times$	$0.3073 \rightarrow 0.3767$

A.7 CARLA Evaluation Issues

Closed-loop evaluation of VLA policies introduces subtle protocol hazards that can systematically distort benchmark comparisons if unaddressed. We highlight three key methodological insights and sanitization procedures:

Infrastructure Exceptions vs. Execution Failures. Out-of-memory (OOM) exceptions during agent initialization can be erroneously recorded by standard evaluation harnesses as zero-score driving failures (DS $= 0.0$). Because memory footprint scales with model context and autoregressive sequence length, this artifact selectively penalizes trajectory-native architectures while leaving lower-capacity baselines unaffected. Proper evaluation hygiene requires explicitly isolating process-level setup exceptions from true closed-loop driving failures to avoid artificial policy suppression.

Inference Latency Bias in Fixed-Timeout Benchmarking. Closed-loop benchmarks typically enforce a fixed wall-clock timeout per route. However, when comparing models with differing per-step inference costs—such as multi-token autoregressive generation versus single-pass continuous heads—a fixed wall-clock budget truncates the maximum simulation time available to slower policies. In environments like CARLA, where specific infractions (e.g., "vehicle blocked") require a minimum duration of continuous immobility to trigger, slower models are killed prematurely before terminal event detectors fire. To eliminate this bias, wall-clock timeouts must be scaled proportionally to per-tick execution latency, ensuring all policies can reach terminal simulation outcomes.

Control Frequency Protocol Alignment (10 Hz vs. 20 Hz). While our evaluation harness executes CARLA at a 10 Hz control rate, standard Bench2Drive protocols default to 20 Hz. To verify that control frequency does not confound relative model performance, we evaluated official baseline weights across both frequencies under identical controllers. The resulting paired score shift ($\Delta = +5.21$ DS, 95% CI ± 12.4) falls within the suite's minimum detectable gap (9.53 DS), demonstrating that frequency variations do not shift relative policy rankings when control primitives and vehicle wrappers remain strictly matched.

A.8 CARLA Closed-Loop Controller Audit

K-means trajectory execution in standard evaluation harnesses suffers from a start-from-rest controller defect: the baseline PID controller derives target speed from the initial 0.5 s waypoint segment, causing accelerating vehicles to be misclassified as stationary and permanently braked (braking expert trajectories 54.8% of the time in pull-away regimes). Replacing this with time-aware speed mapping resolves the deadlock (+18.60 DS shift; 90% attributed to deadlock recovery, 10% to driving refinement). Under fair time-aware decoding, the headline performance gap between CTAR and K-means remains +39.70 DS at matched steps (4.1× the Minimum Detectable Gap of 9.8 DS).

A.9 NAVSIM Codec Reconstruction and Sub-Channel Dynamics

Reconstruction Ceiling Decomposition. On NAVSIM v2, the step-10k OAT codec exhibits a 1.1821 m ADE reconstruction error. A controlled 4-arm codec ablation (Table 8) reveals that this error is driven by:

- **Training Schedule Weighting (67%):** Cosine decay of the codec loss weight ($5.0 \rightarrow 0.25$) under joint optimization starves latent code learning. Isolated codec training reduces ADE to 0.396 m.
- **Normalizer Grid Alignment (8%):** Re-fitting the normalization bounds to empirical label distributions lowers ADE to 0.299 m.
- **Codebook Capacity Constraint (0%):** Expanding capacity 17× (17,280 FSQ codes) yields no improvement (0.384 m ADE, 13% code utilization), confirming that discrete code capacity is not a binding bottleneck.
- **Architectural Depth Floor (≈ 0.25 m):** Residual error reflects encoder-decoder layer depth constraints across short horizons.

Table 7: **Model-free reconstruction ceiling** on `navhard` stage-one windows ($n = 450$). *Continuous heading error reflects finite-difference tangent reconstruction.

Action Interface	Capacity (bits)	ADE (m)	FDE (m)	AHE (rad)	FHE (rad)
`joint_oat` ($32 \times \log_2 1000$)	≈ 319	1.1821	1.6974	0.0445	0.1012
`vlm_kdisk` ($8 \times \log_2 2048$)	88	**0.0540**	**0.0764**	**0.0097**	**0.0121**
`path_speed` (Continuous xy)	n/a	0.0000	0.0000	0.0441*	0.0414*

Table 8: **Codec retrain ablation and capacity scaling** (20k-step matched training budget).

Variant	Configuration	ADE / FDE (`stage-1`)	ADE / FDE (`full`)
Deployed v2 Codec	Joint objective (step-10k snapshot)	1.182 / 1.697	0.914 / 1.194
Arm 1 (Control)	Base grid + MSE loss	0.396 / 0.534	0.366 / 0.449
Arm 2	Base grid + Weighted MSE loss	0.405 / 0.539	—
Arm 3 (Best)	**Refit grid + MSE loss**	**0.299 / 0.427**	**0.276 / 0.371**
Arm 4	Refit grid + Weighted MSE loss	0.338 / 0.473	—
Capacity Probe	17,280 FSQ codes (17×), Base grid	0.384 / 0.474	0.360 / —
`vlm_kdisk`	Frozen K-means baseline	**0.054 / 0.076**	—

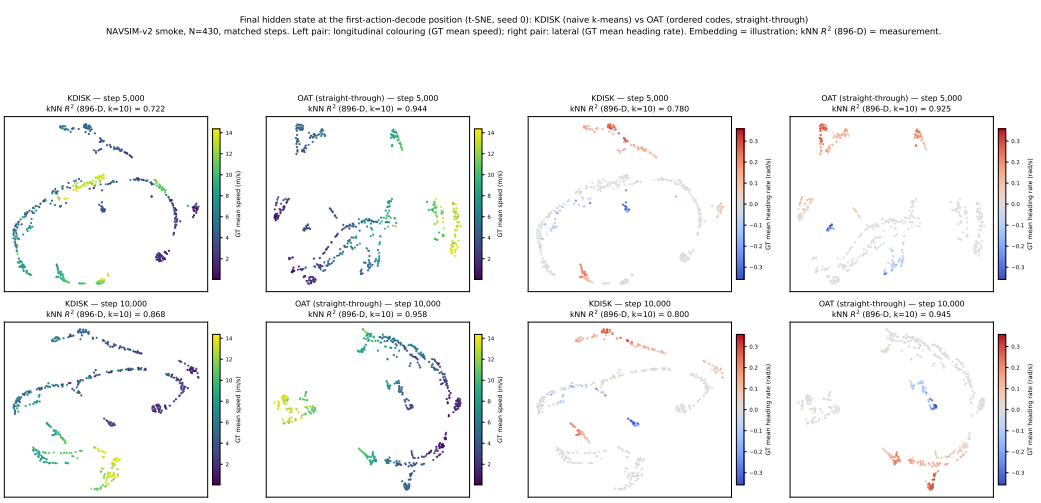


Figure 7: **Latent representation structure across action interfaces.** t-SNE visualization and 10-fold cross-validated k-NN ($k = 10$) regression R^2 scores computed on the final decoder hidden state h prior to logit projection ($N = 430$ anchors). OAT hidden states exhibit significantly stronger linear and non-linear organization relative to continuous physical ground truth across speed ($R^2 = 0.958$ vs. 0.868), longitudinal acceleration (0.486 vs. 0.154), heading rate (0.945 vs. 0.800), and curvature (0.936 vs. 0.763).

Directional Error Diagnostics. Decomposing codec reconstruction errors shows that OAT errors are along-track dominated (along-to-cross track error ratio of 2.1–3.1), inducing a 6.4–13.6× jerk inflation while keeping lateral acceleration at $1.0\times$. Consequently, closed-loop training rapidly resolves lateral Drivable Area Compliance (DAC reaches cohort-best 0.853 at 70k steps), while residual score deficits remain strictly localized to comfort metrics (two-frame extended comfort of 0.373 vs. 0.747 for continuous regression).

A.9.1 Route Conditioning and Causal Redundancy

Evaluating route command responsiveness by intervening on the `driving_command` prompt input ($n = 430$ anchors) reveals a median trajectory shift of only 0.000–0.046 m against a 16 m intent scale. Linear probes trained on the VLM's vision-driven trajectory predictions recover the high-level routing command with **86.2% balanced 3-class accuracy** (ground-truth future ceiling: 88.3%). Because front-camera visual features almost entirely determine the route branch (achieving 97.6% left-turn and 100.0% right-turn execution from vision alone), the model naturally down-weights high-level prompt conditioning due to visual input redundancy.