MINDVLA-U1: VLA BEATS VA WITH UNIFIED STREAMING ARCHITECTURE FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous driving has progressed from modular pipelines toward end-to-end unification, and Vision-Language-Action (VLA) models are a natural extension of this journey beyond Vision-to-Action (VA). In practice, driving VLAs have often trailed VA on planning quality, suggesting that the difficulty lies in the interface through which semantic reasoning, temporal context, and continuous control are combined. We argue that this gap reflects how VLA has been built — as isolated subtask improvements that fail to compose into coherent driving capabilities — rather than what VLA is. We present **MindVLA-U1**, the first unified streaming VLA architecture for autonomous driving. A unified VLM backbone produces autoregressive (AR) language tokens (optional) and flow-matching (diffusion-style) continuous action trajectories in a single forward pass over one shared representation, *preserving the natural output form of each modality*. A full streaming design processes the driving video framewise rather than as fixed video-action chunks under costly temporal VLM modeling. Planned trajectories evolve smoothly across frames while a learned streaming memory channel carries temporal context and updates. The unified architecture enables fast/slow systems on dense/sparse *Mixture-of-Transformers* (MoT) backbones via flexible self-attention context management, and exposes a measurable language-control path for action: a language-predicted driving intent steers the action diffusion via classifier-free guidance (CFG), turning language-side intent into control signals for continuous action planning. On the long-tail WOD-E2E benchmark, MindVLA-U1 *surpasses experienced human drivers for the first time* (8.20 RFS vs. 8.13 GT RFS) with only 2 diffusion steps, achieves state-of-the-art planning ADEs over prior VA/VLA methods by large margins, and matches VA-class latency ($\sim$16 FPS vs. RAP's $\sim$18 FPS at matched $\sim$1B scale) while preserving natural language interfaces for human–vehicle interaction.

1 INTRODUCTION

Driving, at its core, is two things at once — a continuous act of physical control, and a continuous act of understanding. Most of it happens by reflex: the routine lane changes, the gentle braking, the thousand small adjustments that a skilled driver makes without thinking. But the moments that separate competent driving from merely adequate driving are the moments when reflex is not enough — when something in the world demands interpretation, and the correct response depends on knowing what that something means. A system that drives well must do both, and must do them within a single coherent architecture.

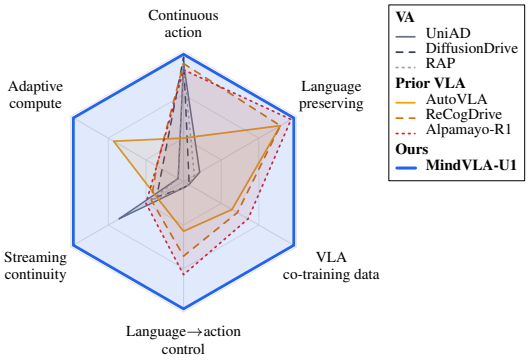


Figure 1: **AD capability radar**

Vision-to-Action (VA) models (Hu et al., 2023; Jiang et al., 2023; Chen et al., 2024a; Sun et al., 2025; Zheng et al., 2024; Weng et al., 2024; Li et al., 2024b; Chitta et al., 2022; Liao et al., 2025; Zheng et al., 2025; Feng et al., 2025) represent the state of the art on the first half of this problem. They map sensor input to trajectories with centimeter-level precision, dominate planning benchmarks, and produce genuinely useful driving within the distribution of scenarios they have seen. But the semantic

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.

structure behind their behavior remains implicit, encoded in latent features tied to the co-occurrence patterns of the training distribution. Such representations can support strong pattern-conditioned action, but they are not explicit, inspectable, or reliably compositional when appearance, category, intent, and required response no longer compose in familiar ways. Vision-Language-Action (VLA) models (Shao et al., 2024; Tian et al., 2024; Jiang et al., 2024; Sima et al., 2024; Hwang et al., 2024; Wang et al., 2025b; Chi et al., 2025b; Zhou et al., 2025; Zeng et al., 2025a; Yuan et al., 2025; Rowe et al., 2025; Fu et al., 2025; Renz et al., 2025; Zhang et al., 2025b; Luo et al., 2025; Li et al., 2025b;a; Wang et al., 2025c; Peng et al., 2025; Huang et al., 2026) are a natural architecture for systems that must do both. A Vision-Language Model (VLM) backbone (Chen et al., 2024b; Yang et al., 2024; 2025; Qwen Team, 2026; Guo et al., 2025) provides the explicit representations VA lacks: compositional, open-vocabulary understanding grounded in language. In principle, this is the missing half of the driving problem. In practice, driving VLAs have often trailed VA on planning quality, which has encouraged the view that language and planning precision are in tension, and that adding understanding costs control.

We argue that this view confuses a design failure with a paradigm limitation. Prior driving VLAs share a consistent set of interface-level shortcomings the framework itself does not impose. First, the action interface is mismatched to control precision (Sima et al., 2024; Hwang et al., 2024; Wang et al., 2025b; Chi et al., 2025b; Zhou et al., 2025; Zeng et al., 2025a; Yuan et al., 2025; Rowe et al., 2025). Some designs decode trajectories as discrete numerical tokens through the language head: VLMs can describe rough positions, not high-precision coordinates, so the language stream imposes a precision floor below what physical control demands and exposes action to language-style hallucination. Others run separate action experts (Black et al., 2024; Intelligence et al., 2025) that read VLM features through cross-attention (Tian et al., 2024; Jiang et al., 2024; Li et al., 2025b): action tokens never enter the VLM's self-attention, so the VLM representation is built without action context. In driving, where centimeter-level control depends on tight VLM-action coupling, this routing reduces the VLM to a feature encoder feeding a VA-style head and forfeits much of the VLM's representational potential. Second, temporal modeling is short-window or chunk-scoped (Zhou et al., 2025; Yuan et al., 2025; Rowe et al., 2025; Renz et al., 2025; Luo et al., 2025; Li et al., 2025b; Wang et al., 2025c; Peng et al., 2025; Huang et al., 2026): because the planner predicts fixed-length *action chunks*, the VLM is asked to absorb temporal context directly from multi-frame inputs, but multi-view driving video carries heavy token redundancy, efficient temporal VLM modeling remains an open problem (Nguyen, 2026), and chunked output creates discontinuities at chunk boundaries. Third, language has no explicit path into action (Shao et al., 2024; Zeng et al., 2025a; Fu et al., 2025; Zhang et al., 2025b; Li et al., 2025a). Most driving VLAs initialize from a pretrained VLM but feed only templated driving commands at inference, collapsing the language capability into a fixed prefix and effectively running the system as VA with VLM weights. The root cause is the absence of driving-specific language–action data: real driving logs lack language supervision, and generic VQA data lacks planning signal. Reported planning gains with language enabled are therefore asserted from the existence of a language head, not demonstrated through a measured language-to-action route. These are interface failures, not paradigm failures, and together they leave the central VLA question open: how to combine semantic understanding, temporal context, and continuous control inside one model without sacrificing precision.

So what makes for a good VLA system for autonomous driving? To address driving as a whole, it should derive its interface from the task rather than from inherited VLM conventions. Driving requires centimeter-level physical control, so actions should remain continuous. The long tail requires open-vocabulary and compositional semantic representations, so the model needs an explicit language-grounded representation rather than only implicit visual features. VLA's promise depends on language knowledge becoming usable by action, so the language pathway must have a measurable route into continuous planning rather than merely sharing a backbone. Driving arrives as a framewise stream, not as a sequence of fixed-length chunks, and the temporal interface must reflect this efficiently rather than forcing the VLM to attend over redundant multi-view fixed-frame videos at constant intervals. Finally, routine and difficult moments impose different compute demands. A common criticism of VLA is that adding a VLM backbone fundamentally increases planning cost by a large margin; we view this as another interface-level misinterpretation — a single architecture should admit both fast and slow modes depending on what the scene requires (Figure 1). Many of these requirements have never been addressed in previous VA/VLAs, and jointly satisfying all of them inside a single VLA is structurally difficult — they pull in different architectural directions — so the capabilities that emerge from joint satisfaction have not been demonstrated by any prior driving VLA: smooth long-horizon planning, controllable language-driven action generation, and fast/slow execution within one model without losing planning quality while preserving the language interface.

We present **MindVLA-U1**, the first unified streaming VLA architecture to address them all. The core of MindVLA-U1 is a *unified shared backbone for scene understanding and action generation*: optional autoregressive (AR) language tokens and flow-matching (Ho et al., 2020; Esser et al., 2024; Liu et al., 2022) (diffusion-style) continuous action trajectories are produced from one shared representation in a single forward pass, preserving the natural output form of each modality — action remains continuous, language remains explicit, and both modalities share the same weights. The backbone runs under a full *streaming paradigm*: each step processes only the current multi-view frame rather than fixed video-action chunks under costly temporal VLM modeling, while a learned streaming memory channel propagates compact visual memory across frames and updates along whole streams, so planned trajectories evolve smoothly framewise without the redundant multi-frame attention that prior VLAs rely on. Through flexible self-attention context management, the unified architecture admits multiple inference modes (language-then-action, action-then-language, action-only) that give the dense VLA backbone fast/slow execution paths from a single checkpoint, and extends naturally to sparse *Mixture-of-Transformers* (MoT) (Vaswani et al., 2017; Liang et al., 2024; Deng et al., 2025) variants for higher action efficiency and extensibility. Finally, for the first time in VLA, MindVLA-U1 exposes a measurable language-to-action bridge: it autoregressively predicts a driving intent token and feeds it as classifier-free guidance (Ho & Salimans, 2022) into the action diffusion, turning language-side intent into a control signal for continuous trajectory generation rather than a capability merely asserted by the presence of VLM pretrained weights. Contributions are summarized as follows:

1. **The first unified shared VLA architecture for autonomous driving.** A single VLM backbone jointly performs scene understanding and continuous action generation, preserving action precision and language capability with explicit language-to-action controllability and flexible fast/slow execution on dense / sparse MoT backbones.
2. **Streaming paradigm with efficient temporal modeling.** A framewise streaming paradigm with efficient memory propagation that supports long-horizon prediction, eliminates action chunking, and provides a computation-efficient temporal modeling scheme for VLA.
3. **State-of-the-art performance on WOD-E2E** (Xu et al., 2025)**.** MindVLA-U1 surpasses experienced human drivers for the first time (8.20 RFS vs. 8.13 GT RFS) with only 2 diffusion steps, achieving state-of-the-art planning ADEs over all prior VA/VLA methods by large margins while preserving a natural-language interface for human–vehicle interaction.

2 METHOD

MindVLA-U1 combines a *unified shared backbone* (§2.1) that jointly performs scene understanding and continuous action generation in a single forward pass over one representation, with a *streaming memory* (§2.2) paradigm that propagates a compact memory channel across frames in whole driving sessions rather than re-attending to multi-frame video chunks (Figure 2). To let language guide continuous action measurably and to match compute to scene demand, we further develop a language-to-action route and a fast/slow execution scheme that runs on dense or sparse MoT variants of this shared backbone.

2.1 UNIFIED SHARED BACKBONE

MindVLA-U1 keeps language autoregressive and action continuous: a single VLM backbone jointly processes vision, ego-state, language, memory, and noisy action tokens through the same self-attention and FFN weights at every layer. Noisy action tokens flow through every transformer layer (Vaswani et al., 2017) of the backbone exactly like vision and language tokens with no separate computation path. Gradients from both training objectives flow through every layer back to all token modalities; A language LM head decodes language tokens and a thin flow-matching MLP reads out the action velocity field at action positions, both in a single forward pass (Figure 2).

Joint AR + Flow Matching on Shared VLM backbones. For a single frame-step with multi-view vision tokens $\mathbf{I}$, ego-state history $\mathbf{e}$, language query $\mathbf{q}$, language answer tokens $\mathbf{a} = (a_1, \ldots, a_L)$, and a length-$N$ sequence of noisy action tokens $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,N})$, the shared backbone f_θ runs all tokens through every transformer layer in a single forward pass. The prefix $(\mathbf{I}, \mathbf{e}, \mathbf{q}, \mathbf{a})$ is causally masked and the action tokens are bidirectionally fully visible. Taking language-then-action path as example, the LM head at language position $l-1$ sees only $(\mathbf{I}, \mathbf{e}, \mathbf{q}, a_{<l})$, while the action MLP at any action position sees the full prefix $(\mathbf{I}, \mathbf{e}, \mathbf{q}, \mathbf{a})$ and all N noisy action tokens. The LM head produces

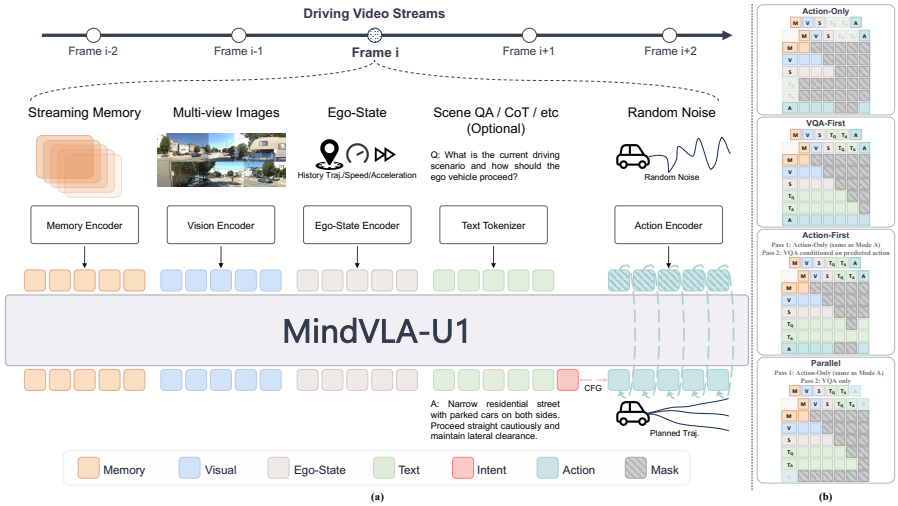


Figure 2: **Overview of MindVLA-U1.** A unified shared backbone decodes language and continuous action from one hidden state in a single forward pass (§2.1); a streaming memory channel carries temporal context across frames, motion-aligned on read and updated after the pass (§2.2).

the autoregressive loss over answer tokens (*e.g.*, scene QAs):

$$\mathcal{L}_{\mathrm{AR}} = -\sum_{l=1}^{L} \log p_\theta(a_l \mid \mathbf{I}, \mathbf{e}, \mathbf{q}, a_{<l}) \,. \tag{1}$$

The action MLP at action positions predicts a velocity field for flow matching. Given ground-truth waypoints $\mathbf{x}_0 = \mathbf{w}$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, the noisy action tokens take the form $\mathbf{x}_t = t\boldsymbol{\epsilon} + (1-t)\mathbf{x}_0$ at $t \sim \mathrm{Beta}(1.5, 1.0)$ under the convention $t{=}0$ is data and $t{=}1$ is noise (Ho et al., 2020; Esser et al., 2024; Liu et al., 2022), giving

$$\mathcal{L}_{\mathrm{FM}} = \left\| v_\theta(\mathbf{x}_t, t;\, \mathbf{I}, \mathbf{e}, \mathbf{q}, \mathbf{a}) - (\boldsymbol{\epsilon} - \mathbf{x}_0) \right\|^2, \tag{2}$$

where v_θ is the action MLP's velocity prediction. The training objective $\mathcal{L} = \mathcal{L}_{\mathrm{AR}} + \mathcal{L}_{\mathrm{FM}}$, and at inference trajectories are recovered by ODE integration (Song et al., 2020) over v_θ.

Training the discrete AR cross-entropy and the continuous flow-matching MSE on shared backbone parameters is stable in practice because the two losses apply at *different output positions* through *separate readout heads*: they share gradients but not targets, and the scene-grounded features each loss needs compose rather than compete. This shared-backbone design separates MindVLA-U1 from three prior VLA patterns: *discrete-token action VLAs* (full-autoregressive (Zhou et al., 2025) or block-diffusion-style (Ma et al., 2025)) decode trajectories as discrete tokens through the language head and inherit a tokenizer-imposed precision floor; *discrete tokens with downstream trajectory decoding* (*e.g.*, Alpamayo-R1 (Wang et al., 2025c)) recovers continuous action from those discrete tokens via a separate decoder, leaving action generation outside the backbone forward pass; and *VLM with separate action experts* (*e.g.*, pi-0 (Black et al., 2024)) routes action through a cross-attention expert with its own attention/FFN, so VLM features are built without action context. Action of our MindVLA-U1 is decoded continuously inside the shared self-attention.

Language-to-Action Bridge via Intent-CFG. Intent is intuitively a natural and compact bridge from language reasoning to continuous action. No matter how complex the situation, after sufficient scene analysis and reasoning a human driver settles on a single driving intent (go straight, change lane, yield) and acts on it. MindVLA-U1 implements this bridge inside the VLM forward pass: the language head is supervised to predict the current-scene intent label z (*e.g.*, Left, Right, Go Straight) via standard next-token prediction, and the predicted intent token is embedded and added to the action MLP's time embedding before velocity prediction. During training, the conditioning intent is occasionally replaced by an unconditional token $\emptyset$ (CFG dropout) (Ho & Salimans, 2022), so the same action tokens learn both conditional and unconditional velocity fields. At inference, MindVLA-U1 decodes an intent token through the language head and runs two backbone forward passes — conditional on z and on $\emptyset$ — mixing the velocity fields by guidance scale s:

$$v_{\mathrm{cfg}}(\mathbf{x}_t, t) = v_\theta(\mathbf{x}_t, t;\, \emptyset) + s\left(v_\theta(\mathbf{x}_t, t;\, z) - v_\theta(\mathbf{x}_t, t;\, \emptyset)\right), \tag{3}$$

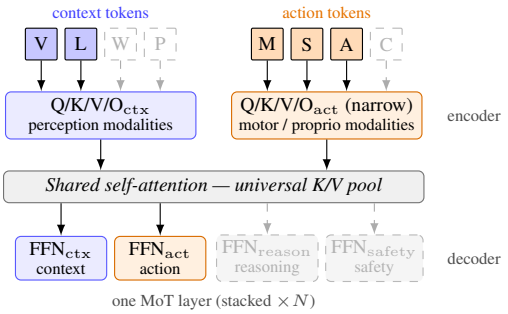


Figure 3: **Fast/Slow systems on Sparse MoT.** Per-modality encoders feed a shared self-attention, per-functionality decoders specialize the outputs. (Dashed: extension slots).

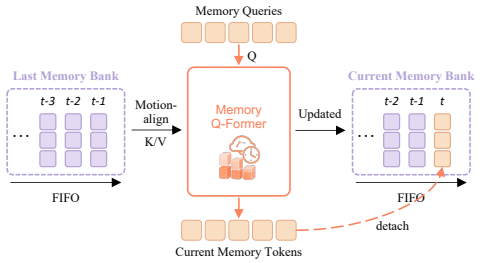


Figure 4: **Streaming Memory.** FIFO memory channel with motion-aligned and refreshed history tokens; gradients propagate across stream.

where $v_\theta(\cdot\,; z)$ extends the velocity field of Eq. 2 with intent conditioning z (the prefix $\mathbf{I}, \mathbf{e}, \mathbf{q}, \mathbf{a}$ is suppressed). This language-to-action controllability inherits from MindVLA-U1's end-to-end, unified shared VLA architecture design: discrete-token action heads have no continuous velocity field for CFG to mix. Post-hoc trajectory decoders or separate action experts detach the gradients between intent prediction and action denoising that U1 maintains.

Fast/Slow Systems. MindVLA-U1's unified shared backbone enables fast/slow systems easily as self-attention context management — no extra modules, gating heads, or runtime selectors are required. Four orderings cover the design space via attention-mask composition alone (Figure 2): `vqa_first/only` decode language first and feeds the answer back as conditioning for action denoising, while `action_first/only` denoises action first and conditions language on the predicted trajectory. These modes serve complementary purposes: `vqa_first` enhances action diffusion — scene QA enriches the action-side context with structured semantic features, and CoT-style (Wei et al., 2022) reasoning can run before the trajectory is committed — while `action_first` produces a natural-language commentary on the just-predicted trajectory, useful in safety-critical scenarios where users need to inspect why a particular maneuver was chosen. The fast path `action_first` drops language entirely, giving a strictly shorter forward pass (VA systems). All orderings are sampled at training to enable flexible inference mode during deployment under one set of weights, which empirically preserves planning quality across modes showing that the VLM backbone does not impose a cost on action even when language is present. The same attention-mask mechanism extends naturally to additional functional experts (reasoning, safety, world model) beyond the language and action heads instantiated here.

The same backbone admits a sparse *Mixture-of-Transformers (MoT)* variant (Liang et al., 2024; Deng et al., 2025) for higher action efficiency and extensibility (Figure 3). The MoT splits each layer into two parallel expert groups — a context group serving visual and language tokens, and an action group serving memory, ego-state, and action tokens — joined by a shared self-attention pool so every query attends over both groups' representations in one forward pass. The two-group structure scaffolds future extensions: new perceptual modalities or cognitive functions enter as additional context-side token-roles, and behavior-style or control heads as additional action-side token-roles. We develop this into a foundation-architecture vision applicable beyond driving in §F.2; implementation details for the deployed two-group instance are in §F.1.

2.2 Streaming Paradigm

MindVLA-U1 processes driving as a framewise stream: at frame i, the VLM consumes only the current multi-view frame — which can arrive at flexible, even non-uniform intervals — plus a compact memory feature $\mathbf{m}_i$ read from a bounded First-In-First-Out (FIFO) memory channel that stores per-frame memory tokens (each a compressed summary of a past frame's backbone state, not raw visual tokens; Figure 4). Historical context propagates through the FIFO channel rather than via repeated multi-frame attention, so planning evolves smoothly across long driving sessions at bounded per-frame cost, without fixed action chunks or redundant temporal-VLM input. Because each token set was written under an earlier ego pose, it is motion-aligned to the current ego frame on read, keeping memory spatially consistent with the current viewpoint as the vehicle moves. The streaming schedule (frames processed in order with channel state carried forward) and the FIFO channel itself are separable components.

Streaming memory updating. After the frame-i forward pass, a Q-Former-style (Li et al., 2023) propagation transformer cross-attends learnable query tokens to the backbone outputs to compute a fresh set of compressed memory tokens, which is appended to the FIFO channel; the oldest set is evicted once the token budget is exceeded, and the first-frame forward simply receives an empty channel. Gradients flow through the propagation transformer across frames, so the loss at frame i supervises the memory written at frame $i-1$. This converts the channel from a passive cache into an actively supervised state: memory features are shaped by the same flow-matching and language objectives that supervise current-frame action and language, with no auxiliary reconstruction loss. The same read–forward–write loop runs at training and inference, so streaming behavior at deployment matches what was trained. Qualitative streaming examples — per-frame inference smoothness and long-horizon multi-clip stitches — are shown in §E.3.

3 Experimental Results

3.1 Dataset, Benchmark, and Implementation Details

Dataset, benchmark, and VLA co-training data. We train and evaluate on *Waymo Open Dataset End-to-End (WOD-E2E)* (Xu et al., 2025): 4,021 ∼20-second driving segments captured by an 8-camera 360° rig, covering long-tail scenarios (event frequency <0.003%). The official split provides 2,037 training and 479 validation sequences; the test split releases the first 12 s of each segment and holds the final 8 s for evaluation. We provide a richer breakdown alongside the official WOD-E2E metrics. For language quality, we report BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) on the VQA answer head. For trajectory quality, we report: (i) the *Rater Feedback Score (RFS)*, in which each predicted trajectory is matched to the closest of three human-annotated reference trajectories (scored in $[0, 10]$), with trajectories inside a speed-scaled trust region receiving the rater's score and those outside decaying exponentially to a floor of 4; (ii) RFS-matched ADE at 3/5 s, computed against the rater-matched reference; and (iii) RFS-GT ADE at 3/5 s, computed against the logged ground-truth trajectory — the convention used by external methods and the official challenge leaderboard. For the long-horizon streaming setting that MindVLA-U1 targets, we additionally report sequence-level ADE at 3/5/10/15/20/25 s on the validation split.

Real driving logs provide trajectories but no language supervision; *MindLabel* closes this gap by generating scene-grounded VQA and fine-grained intent annotations on the same frames. Across WOD-E2E, it covers ∼18.8K clips independently annotated by Qwen3-VL and Qwen3.5-Plus (pipeline and coverage details in §B). For reported results, we co-train on this VQA, the official WOD-E2E 3-class ground-truth intent label, and the flow-matching action loss in one shared-backbone forward pass (§2.1).

Network architecture and implementation. The MindVLA-U1 architecture is VLM-agnostic: the unified shared backbone, streaming memory, and Intent-CFG bridge are all defined over a generic VLM forward pass, so any modern VLM (*e.g.*, InternVL, Qwen2.5/3/3.5-VL, DeepSeek-R1 (Chen et al., 2024b; Yang et al., 2024; 2025; Qwen Team, 2026; Guo et al., 2025)) can serve as the backbone. We report main results on Qwen3-VL-2B. The vision encoder is frozen; the visual merger, language model, ego-history encoders, the streaming memory module, and the lightweight flow-matching action head are jointly trained. The action head predicts position, velocity, and acceleration over a 5-second horizon at 4 Hz with 2 Euler integration steps at inference. End-to-end sequence-streaming training takes approximately 7 hours on 8×H200 GPUs for 50 BF16 epochs with DeepSpeed ZeRO-2 (Rasley et al., 2020); SE(2) pose-chain preprocessing stitches per-clip local frames into one global frame for sequence-level streaming training (§E.2). Full architectural dimensions, optimizer hyperparameters, and flow-matching component weights are in §C.1.

3.2 Main Results

On WOD-E2E, MindVLA-U1 closes the VLA–VA planning gap. On the official test split, MindVLA-U1 + RL leads in RFS (**7.87**) and RFS-GT ADE (**1.09**/**2.66** m), ahead of the closest prior VLA, dVLM-AD (1.29/3.02). This gain is architectural, not RL-dependent: without RL or external AD pretraining, MindVLA-U1 + Intent-CFG reaches 7.92 RFS on val, matching the strongest pure-VA system, RAP-DINO (7.91), while MindVLA-U1 alone ranks second on test (7.77, 1.16/2.67 m) and beats all prior VLA and VA methods on ADE 3s/5s.

Unlike every previous VLA in our comparison, which discards or does not use its language head after trajectory decoding ("–" in Table 1) and thus effectively runs as VA with VLM weights, MindVLA-U1

[1] An improved checkpoint reaches RFS **8.00** (RFS-GT ADE 0.86/2.06, RFS-matched ADE 0.47/1.06, L B/R 0.31/0.52); the table retains 7.92 so Tables 3 and 4 use the same model.

Table 1: WOD-E2E *validation split*. Methods grouped as VA / VLA / MindVLA-U1 (Ours), with the real-human-driver reference (logged GT trajectory scored under the same RFS metric) listed for comparison. **AD-PT**: external autonomous-driving pretraining (✓ yes, × no, ? not reported). *L B/R*: VQA BLEU-4 / ROUGE-L. **Bold** / underline = best / second-best per column among VLA + Ours.

Method	Backbone	AD-PT	L B/R↑	RFS-GT ADE 3/5↓	RFS-matched ADE 3/5↓	RFS↑
VA Methods						
VAD (Jiang et al., 2023)	ImageNet vision encoder (Dosovitskiy et al., 2020)	×	–	3.19 / 5.81	–	4.45
UniAD (Hu et al., 2023)	ImageNet ViT (Dosovitskiy et al., 2020)	×	–	6.50 / 10.81	–	5.78
RAP-DINO (Feng et al., 2025)	DINOv3-H (Siméoni et al., 2025)	✓	–	0.97 / 2.20	–	7.91
VLA Methods (action-only at inference)						
Poutine-Base (Rowe et al., 2025)	Qwen2.5-VL-3B (Yang et al., 2024)	✓	–	1.27 / 2.94	–	8.12
Real human driver (logged trajectory, RFS-scored)						
Human Driver (Rowe et al., 2025)	—	—	—	—	—	8.13
MindVLA-U1 (Ours)						
MindVLA-U1	Qwen3-VL-2B	×	0.30 / 0.49	0.92 / 2.14	0.50 / 1.05	7.83
MindVLA-U1 + Intent-CFG	Qwen3-VL-2B	×	**0.31 / 0.52**	**0.86** / 2.13	**0.47** / 1.07	7.92†
MindVLA-U1 + MoT	Qwen3-VL-2B	×	0.30 / 0.51	0.89 / **2.11**	0.49 / 1.05	7.92
MindVLA-U1 + RL	Qwen3-VL-2B (Yang et al., 2025)	×	–	1.01 / 2.28	0.51 / **1.03**	**8.20**

Table 2: WOD-E2E *official test split*. Conventions as in Table 1.

Method	Backbone	AD-PT	RFS-GT ADE 3/5↓	RFS↑
VA Methods				
ViT-Adapter-GRU	n/r	?	1.20 / 2.70	7.50
Swin-Trajectory	Swin Transformer (Liu et al., 2021)	?	1.21 / 2.81	7.54
DiffusionLTF	DiffusionDrive (Liao et al., 2025)	?	1.36 / 2.98	7.72
UniPlan (DiffusionDrive)	DiffusionDrive (Liao et al., 2025)	?	1.31 / 2.99	7.69
VLA Methods (action-only at inference)				
Open-LLaMA	LLaMA-Vision (Touvron et al., 2023)	?	1.31 / 3.22	7.43
NaiveEMMA	Gemini (Team et al., 2023)	×	1.32 / 3.02	7.53
AutoVLA (Zhou et al., 2025)	Qwen2.5-VL-3B (Yang et al., 2024)	✓	1.35 / 2.96	7.56
dVLM-AD (Ma et al., 2025)	LLaDA-V (You et al., 2025) + SigLIP2 (Tschannen et al., 2025)	✓	1.29 / 3.02	7.63
HMVLM (Wang et al., 2025a)	Qwen2.5-VL-3B (Yang et al., 2024)	?	1.33 / 3.07	7.74
MindVLA-U1 (Ours, Language-preserving)				
MindVLA-U1	Qwen3-VL-2B	×	1.16 / 2.67	7.77
MindVLA-U1 + RL	Qwen3-VL-2B (Yang et al., 2025)	×	**1.09 / 2.66**	**7.87**

reports VQA BLEU-4/ROUGE-L from the same checkpoint that predicts trajectories, preserving both language and planning precision at inference. The remaining val→test RFS gap is distributional rather than architectural: the test split over-represents urgent-stopping and yielding scenes not supervised by the training-time Waymo 3-class intent ontology (*left* / *right* / *straight*; statistics in §B.5). Using richer driving intents to guide action diffusion in complex scenes remains future work. Unless otherwise stated, "MindVLA-U1" refers to the dense configuration in Table 1 (RFS 7.83) in later ablations.

3.3 Language-to-Action Controllability via Intent-CFG

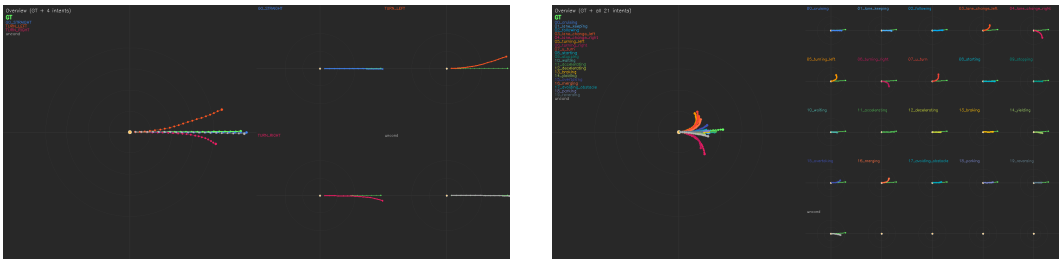
A central claim of the unified shared backbone is that language-side state has a measurable causal route into continuous action. We isolate one capability of this route: predicting a driving intent from the shared representation and using it to steer action diffusion via CFG. All quantitative results use the official WOD-E2E 3-class intent ontology (left, right, and straight). Three conditioning-signal sources are compared against a no-intent baseline — trajectory-derived, GT-supplied, and our next-token-predicted (NTP) variant; embedder implementation details are in §D.

Table 3: Intent-conditioned action diffusion on WOD-E2E val using the official 3-class intent ontology. Variant definitions are described in the surrounding text. † Ours uses a refined NTP-predicted intent embedding; implementation in §D.

Variant	L B/R↑	RFS-GT ADE 3/5↓	RFS-matched ADE 3/5↓	RFS↑
No-intent baseline				
MindVLA-U1 (no intent)	0.30 / 0.49	0.92 / 2.14	0.50 / **1.05**	7.83
Alternative Intent-Injection Mechanisms				
+ Trajectory-derived Intent-CFG	0.30 / 0.49	0.88 / 2.14	0.49 / 1.08	7.81
+ GT-supplied Intent-CFG	0.29 / 0.47	0.89 / 2.15	0.50 / 1.06	7.83
NTP-predicted Intent-CFG (Ours)				
+ NTP-predicted Intent-CFG (Ours)†	**0.31 / 0.52**	**0.86 / 2.13**	**0.47** / 1.07	**7.92**

The three alternatives isolate the conditioning-signal source (Table 3). Trajectory-derived intent (7.81) and GT-supplied 3-class intent (7.83) both essentially match the no-intent baseline (7.83): each exposes a controllable input but does not improve aggregate planning quality. With NTP-predicted intent and a prototype-grounded embedding refinement (§D), MindVLA-U1 + Intent-CFG reaches RFS **7.92** (+0.09 over baseline), the strongest single-axis improvement among MindVLA-U1's component contributions. Intent-CFG is thus both a controllability result — language-side state steers continuous action — and an aggregate planning improvement under the right embedding.

Intent-CFG as a structural multi-modality mechanism. Using the same checkpoint and a representative WOD-E2E frame, Figure 5 varies only the intent condition: the official 3-class ontology



(a) Deployed: WOD-E2E 3-class intent + uncond. (b) Extended: MindLabel 20-class intent + uncond.

Figure 5: **Intent-CFG as a structural multi-modality mechanism.** Per-intent trajectories on the same WOD-E2E frame. Each panel shows a BEV overview with GT (green) on the left and per-intent trajectories on the right: (a) 3-class GT intent; (b) MindLabel's 20-class extension.

and MindLabel's finer 20-class vocabulary produce distinct, semantically consistent trajectory modes, including *u_turn*, *starting*, and *reversing*, while the unconditional input yields a default trajectory close to the GT. Intent-CFG thus exposes an explicit, addressable modal axis for action diffusion, with the finer vocabulary extending controllability beyond left/right/straight.

3.4 Fast/Slow Execution and MoT Design

MindVLA-U1's unified backbone supports sparse MoT routing for fast/slow execution (§2.1). We evaluate the MoT design space here on WOD-E2E val with MindVLA-U1's training schedule (§3.1); end-to-end throughput against a VA reference is reported in the throughput paragraph below.

MoT expert grouping. Table 4 compares two MoT routings with the dense baseline under the same training schedule and AR+FM objective. Both improve over dense (7.83 RFS): *context vs. action*, $(V, L, M, S)+(A)$, peaks at **8.01** RFS, whereas *context vs. proprio+action*, $(V, L)+(M, S, A)$, reaches 7.92 with the lowest RFS-GT ADE at 3s (0.89 m; tied lowest at 5s). We recommend the latter despite its lower headline RFS because keeping memory and ego state with action yields a modality-pure motor group independent of the context group at fast inference; $(V, L, M, S)+(A)$ would instead route memory and ego-state tokens through the context group, undoing MoT's intended throughput separation. Its perceptual → context and motor/proprioceptive → action principle also extends to additional modalities (perception, world model, safety) within the same scaffold, so MoT extends rather than strictly dominates the dense Pareto: one checkpoint supports slow semantic reasoning and fast action-only execution without collapsing planning quality.

Table 4: MoT variant comparison on WOD-E2E val. RFS values are best-of-seeds.

System	L B/R↑	RFS-GT ADE 3/5↓	RFS-matched ADE 3/5↓	RFS↑
Dense Backbone				
MindVLA-U1 (Dense)	0.30 / 0.49	0.92 / 2.14	0.50 / 1.05	7.83
MoT with Different Expert Groupings				
+MoT, {V, L, M, S}+{A} (context vs. action)	**0.31** / 0.49	0.92 / **2.11**	**0.46 / 1.03**	**8.01**
+MoT, {V, L}+{M, S, A} (context vs. proprio+action, Ours)	0.30 / **0.51**	**0.89 / 2.11**	0.49 / 1.05	7.92

Closing the VLA throughput gap to VA. On the same H200, MindVLA-U1 retains language and continuous action while approaching the strongest VA reference: at matched ∼1B scale, its fast path reaches 15.55 FPS versus RAP-DINO's 17.68 FPS (Table 5). At the deployed Qwen3-VL-2B scale, the slow and fast paths share one checkpoint; removing autoregressive answer decoding from the action subgraph—which accounts for ∼94% of slow-path latency, while every other stage costs at most one LM forward—reduces latency from 2,594 to 108 ms while preserving RFS (7.83 vs. 7.82). Although the 1-step fast path reaches 15.92 FPS, its RFS falls to 7.67; the 2-step path (9.26 FPS, 7.82 RFS) is therefore the stronger planning-throughput Pareto point.
The qualitative flow-matching denoising sequence is provided in Appendix C.2, and the broader foundation-architecture interpretation of MoT in Appendix F.2.

3.5 Streaming Memory for Efficient Temporal Modeling

We separately ablate two commitments of the streaming paradigm: *streaming training*, which processes consecutive frames in order so each forward pass sees prior-frame context within the same sample, and an end-to-end updated *memory channel* carrying compact temporal state across stream steps. We also compare the most direct alternative: VLM-side multi-frame video tokens.

Streaming training and memory channel. The two contributions separate cleanly in Table 6: chunk-wise → streaming-no-memory adds 0.04 RFS, and adding memory contributes another 0.10. Memory improves all planning metrics in the streaming setting: rater-matched ADE 5s drops from

Table 5: Inference latency and throughput on WOD-E2E val (bs$=1$, $1\times$ H200). *Top*: Qwen3-VL-2B slow-path (`vqa_first_decoding`) stages; brackets list tokens (V: visual, L: language prompt, Q: VQA question, A: decoded answer, Act: noisy action, M: memory). *Bottom*: slow/fast paths sharing one checkpoint, the RAP-DINO (Feng et al., 2025) VA reference, and the same fast configuration with scale-matched InternVL-2 1B. **Bold**: recommended Pareto points.

Configuration	Latency↓	FPS↑	VRAM↓	RFS↑
Stage-level timing (Qwen3-VL-2B); bracket lists tokens processed per stage.				
Prefix embedding [$V{+}L{+}M$, ≈ 624 tok; vision + past-state + text embed]	∼30 ms	—	—	—
VQA prefilling [$V{+}L{+}M$, ≈ 624 tok; prefill for VQA decoding]	∼40 ms	—	—	—
VQA decoding [A: ≈ 81 tok, autoregressive at 30.4 ms/tok]	∼2,450 ms	—	—	—
Action diffusion, 1 Euler step [Act: ≈ 20 tok appended to $V{+}L{+}M{+}Q{+}A$]	∼38 ms	—	—	—
Action diffusion, 2 Euler steps [Act: ≈ 20 tok appended to $V{+}L{+}M{+}Q{+}A$]	∼74 ms	—	—	—
End-to-end throughput of MindVLA-U1 (Qwen3-VL-2B Dense, No Intent); compared with the strongest VA reference.				
RAP-DINO (Feng et al., 2025) (VA, ∼0.88B)	57 ms	17.68	—	7.91
MindVLA-U1 (InternVL-2 1B, template QA, 2 steps)	64 ms	**15.55**	2,478 MB	**7.78**
MindVLA-U1 (`vqa_first_decoding`, no intent, 2 steps)	2,594 ms	0.39	5,431 MB	7.83
MindVLA-U1 (`action_only`, 2 steps)	103 ms	9.70	5,412 MB	7.74
MindVLA-U1 (`vqa_first_fast`, template QA, 2 steps)	108 ms	**9.26**	5,412 MB	**7.82**
MindVLA-U1 (`vqa_first_fast`, template QA, 1 step)	63 ms	15.92	5,412 MB	7.67

1.14 m to **1.05** m, and sequence ADE falls at every horizon (e.g., 25 s: 1.54 → **1.50** m). The roughly flat RFS-GT ADE 3s is consistent with memory carrying scene-style and behavioral-context cues that affect rater-judged quality more than raw L2 distance to logged GT. This ablation does not isolate gradient-connected from detached-gradient training, which we leave to future work.

Table 6: Streaming training and memory channel ablation on WOD-E2E val. $^\flat$ chunk-wise + VLM-side temporal modeling (Qwen3-VL DeepStack).

Configuration	L B/R↑	RFS-GT ADE 3/5↓	RFS-matched ADE 3/5↓	seq ADE 3/5/10/15/20/25↓	RFS↑
Chunk-wise Training Baseline (single-clip per forward, no streaming memory)					
Chunk-wise, single frame	0.29 / 0.51	0.92 / 2.15	0.52 / 1.13	—	7.69
Chunk-wise + image sequence (4 frames)$^\flat$	**0.31 / 0.53**	1.14 / 2.53	0.63 / 1.27	—	7.61
Streaming Training Variants					
Streaming training, no memory channel	0.31 / 0.49	0.98 / 2.30	0.50 / 1.14	0.48 / 0.98 / 1.03 / 1.22 / 1.33 / 1.54	7.73
MindVLA-U1 (streaming + memory, Ours)	0.30 / 0.49	**0.92 / 2.14**	**0.50 / 1.05**	**0.46 / 0.94 / 0.99 / 1.19 / 1.29 / 1.50**	**7.83**

Temporal VLM modeling vs. streaming memory. To test where temporal context should enter, we compare the single-frame VLM plus compact learned memory (§2.2) with Qwen3-VL DeepStack, which stacks vision tokens from the $K{=}4$ most recent frames at the visual encoder so VLM attention handles history (Table 6, $^\flat$ row). DeepStack slightly underperforms the chunk-wise single-frame baseline (RFS 7.61 vs. 7.69) and does not recover the streaming-memory gain (7.83): multi-view video tokens are heavily redundant, and a generic VLM is not architecturally trained to compress them into a planning-relevant temporal state, whereas the propagation transformer's memory channel is supervised end-to-end and trained to read and write that state.

3.6 RL Post-Training

As a final step toward state-of-the-art planning quality, we post-train the SFT checkpoint with Group Relative Policy Optimization (GRPO) (Guo et al., 2025) using the rater-feedback score (RFS) as the only reward signal — no auxiliary ADE, FDE, or smoothness terms — to test whether the unified streaming interface absorbs a rater-aligned reward without architectural change. RL post-training pushes RFS from 7.83 (SFT init) to **8.20** on the validation split and to a leading **7.87** on the official test split (Tables 1 and 2); the trust-region rate (the share of predicted trajectories falling inside at least one rater's trust region) rises from 66.0% to **73.1**%. RL hyperparameters and ADE-rater trade-offs are in §G, while Appendix B.5 further analyses the WOD-E2E val/test intent-distribution shift.

4 Conclusion

MindVLA-U1 attributes the real-world-driving VLA–VA planning gap not to a paradigm cost of semantic reasoning but to interface design. It closes this gap with a unified shared backbone producing autoregressive language and flow-matching continuous action in one pass over one representation; streaming memory replacing redundant multi-frame VLM modeling and fixed action chunks; an Intent-CFG bridge giving language a measurable causal route into action; and dense/sparse-MoT fast/slow execution recovering VA-class throughput from one checkpoint. With two diffusion steps, the resulting system reaches the highest reported official-test RFS; RL post-training further surpasses experienced human raters on val (8.20 vs. 8.13 GT RFS). The val/test diagnostic (§B.5) identifies a benchmark-level intent shift behind the residual gap, supporting the central thesis: *the real-world long-tail driving VLA–VA gap is interface-level, and the unified interface closes it without sacrificing precision, throughput, or the natural-language interface*.

AI USE STATEMENT

In this work, we used Qwen3-VL and Qwen3.5-Plus as annotation models in MindLabel to generate scene-understanding questions and answers and fine-grained driving-intent annotations (Appendix B). We also used Codex to assist with LaTeX preparation, manuscript organization and drafting, and interpretation and consistency checking of reported results. The VLM backbone of MindVLA-U1 is part of the method itself; its role is described in §2.1. Model-generated annotations can contain errors and are not equivalent to human-validated labels. The authors remain responsible for reviewing AI-assisted material and for the final text, claims, and artifacts of this work.

ETHICS STATEMENT

This work studies trajectory planning using recorded driving data from WOD-E2E. The reported open-loop evaluations do not establish closed-loop safety or readiness for on-vehicle deployment. MindLabel's synthetic annotations may contain errors or reproduce biases, particularly for under-represented driving situations, and should not be treated as verified human judgments. The underlying dataset and pretrained models are third-party resources; their use and any distribution of derived artifacts must respect the original access and licensing conditions and protect personal information. Deployment would require independent safety validation beyond the experiments reported here.

REPRODUCIBILITY STATEMENT

The shared-backbone objectives and streaming procedure are described in §2.1 and §2.2; the benchmark, metrics, and main experimental settings are specified in §3.1. Appendix B describes the MindLabel scene-understanding and intent-annotation pipeline. Further architectural dimensions and optimization settings are listed in §C.1, Intent-CFG conditioning in §D, MoT routing in §F.1, and RL post-training settings in §G.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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

Yandu Chen, Kefan Gu, Yuqing Wen, Yucheng Zhao, Tiancai Wang, and Liqiang Nie. Intentionvla: Generalizable and efficient embodied intention reasoning for human-robot interaction. *arXiv preprint arXiv:2510.07778*, 2025.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pp. 24185–24198, 2024b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025a.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *arXiv preprint arXiv:2505.23757*, 2025b.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Huang Fang, Mengxi Zhang, Heng Dong, Wei Li, Zixuan Wang, Qifeng Zhang, Xueyun Tian, Yucheng Hu, and Hang Li. Robix: A unified model for robot interaction, reasoning and planning. *arXiv preprint arXiv:2509.01106*, 2025.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. *arXiv preprint arXiv:2510.04333*, 2025.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Wenhui Huang, Songyan Zhang, Qihang Huang, Zhidong Wang, Zhiqi Mao, Collister Chua, Zhan Chen, Long Chen, and Chen Lv. Automot: A unified vision-language-action model with asynchronous mixture-of-transformers for end-to-end autonomous driving. *arXiv preprint arXiv:2603.14851*, 2026.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 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.

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 *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European conference on computer vision*, pp. 142–158. Springer, 2024a.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025a.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025b.

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, 2024b.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

Yingzi Ma, Yulong Cao, Wenhao Ding, Shuibai Zhang, Yan Wang, Boris Ivanovic, Ming Jiang, Marco Pavone, and Chaowei Xiao. dvlm-ad: Enhance diffusion vision-language-model for driving via controllable reasoning. *arXiv preprint arXiv:2512.04459*, 2025.

Thong Thanh Nguyen. Video understanding: Through a temporal lens. *arXiv preprint arXiv:2602.00683*, 2026.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Zhenghao Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, et al. Counterfactual vla: Self-reflective vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2512.24426*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11993–12003, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15120–15130, 2024.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

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.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801. IEEE, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Daming Wang, Yuhao Song, Zijian He, Kangliang Chen, Xing Pan, Lu Deng, and Weihao Gu. Hmvlm: Multistage reasoning-enhanced vision-language model for long-tailed driving scenarios. *arXiv preprint arXiv:2506.05883*, 2025a.

Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3621–3631, 2023.

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 computer vision and pattern recognition conference*, pp. 22442–22452, 2025b.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

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

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. Autodrive-r^2: Incentivizing reasoning and self-reflection capacity for vla model in autonomous driving. *arXiv preprint arXiv:2509.01944*, 2025.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025a.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, Xing Wei, and Ning Guo. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *arXiv preprint arXiv:2509.22548*, 2025b.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, et al. 4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration. *arXiv preprint arXiv:2506.22242*, 2025a.

Ruifei Zhang, Junlin Xie, Wei Zhang, Weikai Chen, Xiao Tan, Xiang Wan, and Guanbin Li. Adadrive: Self-adaptive slow-fast system for language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5112–5121, 2025b.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025.

Zewei Zhou, Tianhui Cai, Seth Z 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. *arXiv preprint arXiv:2506.13757*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Related Work

This section surveys VA and VLA methods primarily from the autonomous-driving perspective. Because driving and embodied intelligence share structural questions about action representation, temporal modeling, and language–action coupling despite their different observation and action spaces, the discussion draws on both literatures where their design issues align.

Vision-to-Action (VA) models. End-to-end VA methods (Hu et al., 2023; Jiang et al., 2023; Chen et al., 2024a; Sun et al., 2025; Zheng et al., 2024; Weng et al., 2024; Li et al., 2024b; Chitta et al., 2022; Liao et al., 2025; Zheng et al., 2025; Feng et al., 2025) map camera observations directly to continuous trajectories. Their directness is their strength: action remains continuous, the optimization target stays close to the control objective, and strong VA systems set a high bar for trajectory quality. Their limitation is semantic: the learned representations remain implicit, without a language-mediated interface for inspecting scene concepts, conditioning on intent, or testing whether language-grounded knowledge changes the action. This limitation motivates adding explicit semantic grounding without giving up VA's continuous-control discipline.

Driving Vision-Language-Action (VLA) models. Driving VLA methods (Shao et al., 2024; Tian et al., 2024; Jiang et al., 2024; Sima et al., 2024; Hwang et al., 2024; Wang et al., 2025b; Chi et al., 2025b; Zhou et al., 2025; Zeng et al., 2025a; Yuan et al., 2025; Rowe et al., 2025; Fu et al., 2025; Renz et al., 2025; Li et al., 2025b;a; Wang et al., 2025c) attach VLM-style semantic reasoning to action generation, but often inherit interfaces convenient for language models rather than natural for control. We organize the resulting gaps along action representation, language–action coupling, fast/slow execution, temporal continuity and temporal-token efficiency, driving-native supervision, and real-world evaluation. MindVLA-U1 is positioned as an interface correction across these axes rather than as a single isolated module.

Continuous action and open-vocabulary grounding. Tokenizing trajectories makes action compatible with an autoregressive language decoder but imposes a precision floor on spatial regression (Sima et al., 2024; Hwang et al., 2024; Wang et al., 2025b; Chi et al., 2025b; Zhou et al., 2025; Zeng et al., 2025a; Yuan et al., 2025; Rowe et al., 2025; Zitkovich et al., 2023; Kim et al., 2024). Bypassing language recovers continuous control but gives up the explicit semantic representation that motivates VLA in the first place. Dual-head VLA designs (Tian et al., 2024; Jiang et al., 2024; Li et al., 2025b; Liu et al., 2024; Black et al., 2024; Intelligence et al., 2025) avoid this false choice by sharing representation while preserving modality-native output spaces. MindVLA-U1 follows this principle with autoregressive language and continuous flow-matching action (Chi et al., 2025a; Qu et al., 2025) on one backbone.

Language-action coupling. A recurring under-treated question in VLA-AD is whether the language reasoning produced by the VLM actually contributes to the action — or whether the language head is a costly side-task the action head learns to ignore. In broader robot learning, goal-conditioned and intent-conditioned policies (Zitkovich et al., 2023; Wu et al., 2023; Chen et al., 2025) provide mechanistic precedents for explicit language-to-action routing; in the driving VLA setting, however, an explicit and measurable bridge from language-side state to continuous action has not yet been demonstrated. Prior driving-VLA work typically reports a single planning number with language enabled, without ablating against parallel or action-only variants that isolate the contribution of the causal language-to-action conditioning (Wang et al., 2025c; Peng et al., 2025). The result is that the field cannot distinguish "language helped planning" from "the model with language got better numbers for unrelated reasons."

Fast/Slow Execution System. VLA backbones introduce inference latency incompatible with real-time control, yet most driving moments are routine reflex needing no deliberation — making fast/slow separation a deployment-level necessity. Prior approaches fall into two categories, each with structural limitations. The *cascaded dual-module pipeline* (Tian et al., 2024; Jiang et al., 2024; Li et al., 2025b; Black et al., 2024; Intelligence et al., 2025; Fang et al., 2025) delegates reasoning to a separate VLM whose discrete outputs feed a downstream planner; this severs end-to-end gradient flow and adds serial latency as an unavoidable bottleneck. *Heuristic routing* (Zhang et al., 2025b; Luo et al., 2025; Zhou et al., 2025) uses model inference outputs to decide per frame whether to invoke full deliberation or bypass it; the routing decision lacks distributional robustness and provides no architectural guarantee that the fast path produces valid actions. Deployment-grade fast/slow separation must instead be a structural property of the architecture itself: MoT (Liang et al., 2024;

Deng et al., 2025; Huang et al., 2026) embeds modality-specific attention groups and feed-forward experts directly in the forward pass, making the fast/slow boundary an architectural invariant rather than a runtime heuristic (§2.1).

Action chunking and temporal context modeling. Most published VLA systems predict action chunks and stitch them at clip boundaries, accumulating seam error and breaking the continuity that physical control requires. Their temporal context is also commonly supplied by repeatedly encoding redundant multi-frame VLM inputs, coupling temporal range to growing token and compute costs. Because driving is inherently temporally causal — intent, momentum, and scene evolution carry across observation boundaries — the training paradigm should satisfy three requirements: continuously renewed memory propagation that mirrors the sequential nature of the task, deployment-friendly bounded compute, and end-to-end joint optimization of the memory component with the VLA model. We adopt framewise streaming memory propagation as this paradigm (§2.2). Prior approaches either train on independent chunks without cross-boundary propagation (Zhou et al., 2025; Yuan et al., 2025; Rowe et al., 2025; Renz et al., 2025; Luo et al., 2025; Li et al., 2025b; Wang et al., 2025c; Peng et al., 2025; Huang et al., 2026; Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025), or fail to properly design memory propagation and jointly optimize it with the VLA model (Wang et al., 2023; Li et al., 2024a; Fu et al., 2025; Zhang et al., 2025b; Team et al., 2024; Zhang et al., 2025a; Shi et al., 2025; Qu et al., 2025; Zeng et al., 2025b; Wei et al., 2025) — remaining a suboptimal paradigm choice.

Driving-native VLA supervision. Logged trajectories teach control but not language-grounded scene reasoning: a model trained only to reproduce trajectories receives no explicit supervision for spatial cognition, planning-relevant interpretation, or a vocabulary that distinguishes fine-grained driving intents. MindLabel addresses this gap by binding scene-understanding QA and intent annotations to the same frames used for action supervision.

Simulator-bound evaluation. A practical shortcoming that cuts across the above is the heavy reliance on simulated benchmarks — particularly CARLA (Dosovitskiy et al., 2017) — in academic VLA-AD work. Simulator visual realism, agent behavior, and long-tail composition do not match on-road conditions; closed-loop scores reported on simulator benchmarks routinely fail to predict real-world performance. The field has begun to migrate to real-world benchmarks such as Waymo Open Dataset End-to-End (Sun et al., 2020; Xu et al., 2025), but most published VLA results remain simulator-anchored. MindVLA-U1 is evaluated entirely on real-world WOD-E2E.

B MindLabel Dataset

MindLabel supplies the driving-native language supervision used by MindVLA-U1. Logged trajectories alone provide no explicit supervision for scene cognition: perception of complex layouts, spatiotemporal reasoning about agents, and planning-relevant interpretation. MindLabel therefore generates scene-grounded QA from the same driving frames used for action supervision and separately annotates a fine-grained driving-intent vocabulary. Figure 6 summarizes the two-stage scene-understanding pipeline, and Figure 7 shows its five QA categories on one driving clip.

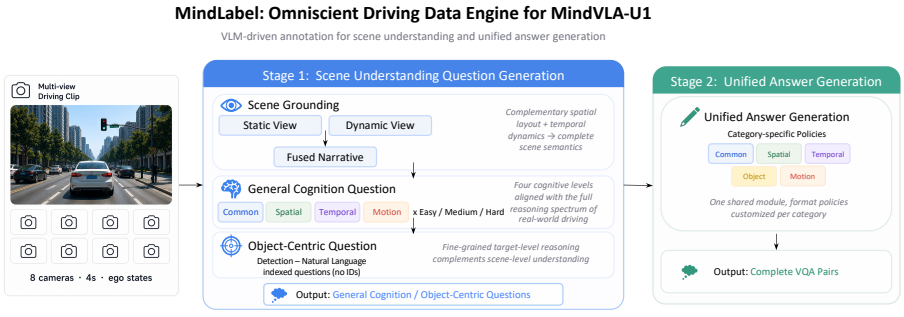


Figure 6: **MindLabel scene-understanding pipeline.** Each 8-camera, 4-second driving clip is grounded into a fused static–dynamic narrative, from which general-cognition and object-centric questions are generated. A shared answer-generation module then produces complete VQA pairs using category-specific policies.

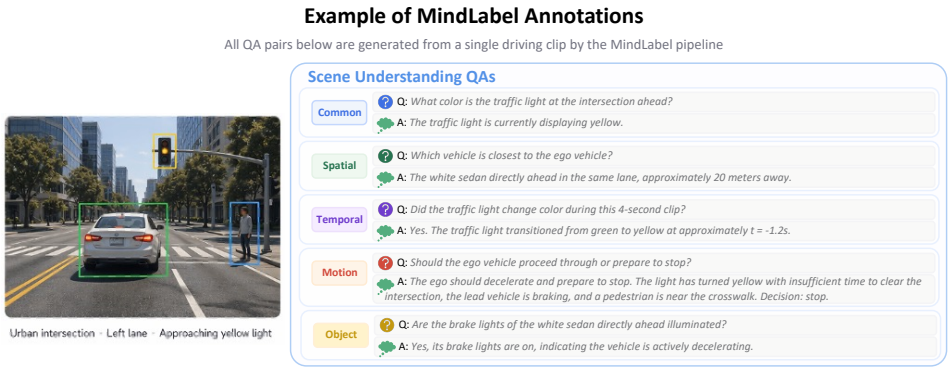


Figure 7: **Example MindLabel annotations from one 8-camera driving clip.** The left panel visualizes one representative camera view; the pipeline generates scene-understanding QA pairs across Common, Spatial, Temporal, Motion, and Object-Centric categories from the multi-view clip.

B.1 Scene Understanding Question Generation

This stage generates structured questions across multiple levels of scene understanding in three steps.

Scene Grounding. The pipeline fuses a *static* view of the most recent frame's spatial layout with a *dynamic* view of the preceding 4-second history window into a single free-form natural-language scene description, intentionally without metric coordinates or structured object lists; precise per-target attributes are handled by Object-Centric Question Generation below.

General Cognition Question Generation. Building on the scene narrative, the pipeline generates questions across four cognitive categories: *Common* (factual scene perception, *e.g.*"*What color is the traffic light ahead?*"), *Spatial* (relative positions and distances), *Temporal* (sequential changes and causal reasoning), and *Motion* (safety analysis and planning decisions, *e.g.*"*Should the ego vehicle proceed, yield, or stop?*"). Each question is independently assigned a difficulty level (Easy/Medium/Hard) and a chain-of-thought flag indicating whether multi-step reasoning is required.

Object-Centric Question Generation. The pipeline performs unified detection across vehicles, pedestrians, cyclists, traffic signs, and signals in the current frame, and generates questions anchored to each detected instance, referencing it through a natural-language description (*e.g.*"*the red truck in the left lane approximately 15 meters ahead*") rather than a detection ID. Question types span object existence, visual attributes, OCR on traffic signs, signal state classification, and counting queries.

B.2 Fine-Grained Intent Annotation

For each frame, MindLabel derives a 20-class intent label from the ground-truth future trajectory. The vocabulary expands the official left/right/straight ontology into longitudinal maneuvers (*e.g.*cruising and decelerating), lateral maneuvers (*e.g.*lane changes and turns), and complex cases (*e.g.*U-turn and emergency stop), exposing a broader and finer-grained annotation space for driving-scene understanding and val/test distribution diagnosis.

B.3 Unified Answer Generation

Scene-understanding questions are answered by a shared module with category-specific formatting policies: *Common* produces concise factual responses, *Spatial* adds explicit units and relative positions, *Temporal* cites time points and causal relationships, and *Motion* concludes with an actionable driving recommendation (*e.g.proceed*, *yield*, *stop*). Questions carrying the chain-of-thought flag receive multi-step inference chains with cited evidence.

B.4 Annotation Coverage

Table 7 reports the total MindLabel scene-understanding annotation scale across the full WOD-E2E benchmark (training + validation + test). Each clip is annotated independently by both backbones (Qwen3-VL and Qwen3.5-Plus), and the VQA totals below aggregate both backbones' outputs on the same clip set. Training and validation clip counts are measured from the labeling runs (8-camera multi-view, 2 Hz over a 4 s window); the test count follows the WOD-E2E protocol of three annotated

clips for each of 1,505 sequences. The retained category counts sum to approximately 3.5M VQA pairs, or approximately 186 pairs per clip across both backbones; category proportions are normalized over these counts. The two backbones contribute roughly equally to this aggregate but not exactly 1:1, since each backbone independently determines how many spatial, temporal, and motion follow-ups to emit per scene. MindVLA-U1's main result consumes the basic scene-grounded VQA stream and the official WOD-E2E 3-class intent label; MindLabel additionally provides the fine-grained 20-class intent vocabulary and chain-of-thought rationales for broader driving-scene analysis.

Table 7: Total MindLabel scene-understanding annotation scale across the full WOD-E2E benchmark (training + validation + test). Each clip is independently annotated by both Qwen3-VL and Qwen3.5-Plus; VQA totals aggregate both backbones' outputs. Category proportions are normalized over the retained category counts.

Statistic	**Training**	**Validation**	**Test**	**Total**
Sequences (WOD-E2E)	2,037	479	1,505	4,021
Annotated clips (per backbone)	10,542	3,771	4,515	18,828
Total scene-understanding VQA pairs	∼1.95M	∼700K	∼835K	**∼ 3.5M**
Common (∼12.3%)	—	—	—	∼430K
Spatial (∼14.9%)	—	—	—	∼520K
Temporal (∼16.0%)	—	—	—	∼560K
Motion (∼16.0%)	—	—	—	∼560K
Object-Centric (∼12.3%)	—	—	—	∼430K
Chain-of-thought, cross-category (∼28.6%)	—	—	—	∼1.0M
Annotation backbones	Qwen3-VL + Qwen3.5-Plus			

B.5 Val/Test Intent Distribution Shift

A central caveat for any WOD-E2E result is that the validation and official test splits are not drawn from the same intent distribution. Applying MindLabel's labels to both splits, Table 8 reports the full 15-intent distribution on val (anchored to RFS-evaluated frames) and test (end-frame protocol). Val is dominated by active-driving intents (*accelerating* 19.5%, *starting* 10.5%), whereas test is dominated by *waiting* (20.2%); *accelerating* falls to 4.1% (−78.9% relative). Other slowdown and yielding categories (*decelerating*, *following*, *yielding*, and lane changes) are also over-represented on test. This shift tracks the val-to-test RFS drop observed across leaderboard methods (MindVLA-U1: −0.24 RFS), reframing the gap as a benchmark-level distribution shift rather than a per-method weakness. MindLabel therefore makes the benchmark itself diagnosable in addition to supervising the model.

Table 8: MindLabel-derived intent distribution on WOD-E2E val and test. Per-intent count n and share (%); sorted by descending val share. **Bold** = dominant intent on each split.

	Val (RFS-anchored)		Test (end-frame)	
Intent	n	%	n	%
accelerating	93	**19.54**	62	4.12
cruising	65	13.66	230	15.29
turning_right	58	12.18	86	5.72
starting	50	10.50	68	4.52
stopping	42	8.82	113	7.51
avoiding_obstacle	38	7.98	37	2.46
waiting	36	7.56	304	**20.21**
turning_left	25	5.25	75	4.99
decelerating	21	4.41	87	5.78
following	13	2.73	84	5.59
lane_change_right	8	1.68	38	2.53
braking	8	1.68	16	1.06
lane_change_left	6	1.26	36	2.39
yielding	5	1.05	35	2.33
u_turn	5	1.05	3	0.20

C Implementation Setup

C.1 Architectural and Optimization Details

This section provides the dimensional and optimization detail deferred from §3.1.

Network architecture. MindVLA-U1 uses Qwen3-VL-2B as the VLM backbone (hidden size H=2048). The vision encoder is frozen; the visual merger, the language model, ego-history encoders, the streaming memory module, and the action head are jointly trained. Ego-history is encoded by three lightweight MLPs (one each for position, velocity, and acceleration) consuming 16 historical states sampled at 2 Hz. The propagation transformer reads/writes N_m=128 memory tokens per frame via 6 cross-attention layers with 16 heads; the FIFO memory channel holds N_g=2 frame-step entries (total capacity $N_g \cdot N_m$=256 tokens). The action head is a 2-layer MLP with SiLU activation that predicts a 6-dimensional output (position + velocity + acceleration) over L_f=20 future waypoints at 4 Hz (5-second horizon), using 2 Euler integration steps at inference.

Implementation details. We optimize with AdamW (lr=10^{-4}, weight decay 0.1, β_1=0.9, β_2=0.999) under a linear-warmup cosine-annealed schedule (warmup 1,000 iterations, $\eta_{\min}$=0.1$\cdot$ $\eta_{\max}$), trained for 50 epochs (§3.1). Mixed-precision BF16 with DeepSpeed ZeRO-2 is used across 8 GPUs. The flow-matching loss applies per-component weights w_{pos}=1.0, w_{vel}=0.5, w_{acc}=0.5. The action representation uses a delta-position channel: future positions are predicted as incremental displacements while velocity and acceleration remain auxiliary absolute channels.

C.2 Flow-Matching Denoising Visualisation

Figure 8 shows the flow-matching action head denoising over 5 Euler steps on a representative WOD-E2E frame: starting from Gaussian noise, the predicted trajectory (blue) is progressively integrated along the learned flow toward the ground-truth corridor (green).

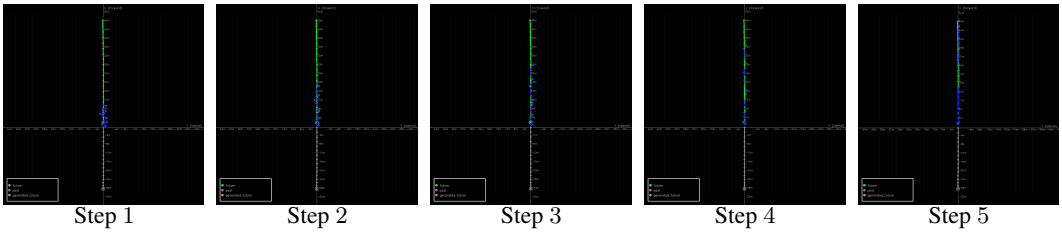


Figure 8: **Flow-matching denoising over 5 Euler steps** on one WOD-E2E frame (BEV: ego forward $+X$, lateral $+Y$) to better demonstrate the denoising process than 2 steps. Green: GT future; gray: past; blue: predicted trajectory after each denoise step. The Gaussian noise input that precedes Step 1 is not shown.

D Intent-CFG: Implementation Details

This section documents the three conditioning sources compared in Table 3 and the CFG mechanism that mixes their conditional and unconditional velocity fields.

Conditioning sources. Three intent signals are evaluated against the no-intent baseline. *Trajectory-derived* extracts an intent class from the GT trajectory geometry. *GT-supplied* uses the WOD-E2E raw intent $z \in \{\text{left}, \text{right}, \text{straight}\}$ label directly — an oracle that decouples CFG-mechanism quality from intent-prediction quality. *NTP-predicted* (the deployed primary) decodes the intent token from the language head via standard next-token prediction on the same scenes that supply the action labels (§2.1), and is paired with the prototype-grounded refinement of the intent embedder referenced in main text.

Embedding and CFG conditioning. An embedding table stores one row per intent class plus a learned unconditional row $\emptyset$. The selected row is projected and added residually to the action MLP's time embedding, preserving the joint AR+FM forward pass of §2.1. At training, the conditioning intent is replaced by $\emptyset$ with probability p_{drop}=0.15, so the same parameters learn both conditional and unconditional velocity fields. At inference, two backbone passes are run — one with the predicted z and one with $\emptyset$ — and their velocity predictions are mixed at every Euler step using Eq. 3 with guidance scale s=1.5.

E STREAMING MEMORY AND LONG-SEQUENCE TRAINING

E.1 ARCHITECTURAL DETAILS

This section gives the architectural details deferred from §2.2. Three coupled components produce the memory feature $\mathbf{m}_i \in \mathbb{R}^{N_m \times H}$ (Figure 4): a FIFO memory channel, a motion-aware modulator, and a propagation transformer.

FIFO memory channel. The memory channel $\mathcal{M}$ is a bounded FIFO buffer holding at most N_g frame-step entries, each consisting of N_m tokens in $\mathbb{R}^H$, for total capacity $N_g \times N_m$ tokens. When full, the oldest entry is evicted, keeping per-step cost constant.

Motion-aware modulation. Each entry in $\mathcal{M}$ is stored in the ego coordinate system of the frame-step that produced it. Before reading, each historical entry j is re-expressed in the current step i's ego coordinate system via the relative SE(2) transform:

$$T_{j \to i} = P_i \cdot P_j^{-1}, \tag{4}$$

where $P_i, P_j \in \mathrm{SE}(2)$ are vehicle poses extracted from ego-states $\mathbf{e}_i, \mathbf{e}_j$. From $T_{j \to i}$ we form a 5-dimensional feature: rotation $(\cos\phi, \sin\phi)$, translation (δ_x, δ_y), and a normalized temporal offset. A lightweight MLP maps this feature to a modulation vector in $\mathbb{R}^H$, added elementwise to all N_m tokens of entry j, yielding the modulated channel contents $\tilde{\mathcal{M}}_i$.

Propagation transformer. N_m learnable query vectors cross-attend to $\tilde{\mathcal{M}}_i$ via a Q-Former-style transformer, producing $\mathbf{m}_i$, which is prepended to the current frame's input sequence. After the backbone forward pass, the same transformer symmetrically compresses the backbone outputs $\mathbf{h}_i$ into N_m tokens that are written back to $\mathcal{M}$ for subsequent frame-steps. Gradients flow through the propagation transformer across stream steps, so step-i losses directly supervise the memory written at step $i-1$ (§2.2). An empirical comparison against the alternative of feeding multi-frame video directly into the VLM is given in §3.5 (Table 6, $^\flat$ row).

E.2 FULL-SEQUENCE POSE RECOVERY FOR STREAMING TRAINING

WOD-E2E releases each driving segment as a sequence of ∼5-second clips, with each clip's trajectory expressed in its own local ego frame. To run streaming training over full segments rather than isolated clips, MindVLA-U1 first recovers a global pose chain that stitches consecutive clips into a single ego-anchored coordinate system, so the streaming memory channel and the per-frame ego-state encoder see consistent positions across the whole segment.

SE(2) pose alignment. At each clip boundary, the relative SE(2) transform between two adjacent clips' ego frames is estimated by aligning the overlapping waypoints (the tail of clip j with the head of clip $j{+}1$) under a rigid 2D rotation + translation. The resulting $T_{(j+1)\to j} \in \mathrm{SE}(2)$ takes coordinates from clip $j{+}1$'s frame into clip j's frame; cumulative composition along the clip chain expresses every clip's local trajectory in a single global frame anchored to the segment's first clip. This is the same SE(2) algebra used inside the streaming memory channel for motion-aware modulation (§E.1, Eq. 4); here it is applied at preprocessing time across clip boundaries.

Recovery quality. Figure 9 visualizes the recovered pose chain on a representative WOD-E2E segment (229 frames). Per-frame alignment residual stays well below ∼0.005 m (mean ∼0.0011 m, 10 inliers per join), the recovered global trajectory traces a geometrically consistent path, and the derived velocity and acceleration profiles are smooth and physically plausible across the full segment. Bottom: four sampled front-view frames with the projected ego trajectory overlaid in red. With this preprocessing in place, MindVLA-U1's streaming forward pass can be trained on full segments, so the FIFO memory channel sees coherent context across many tens of frames and the ego-state encoder consumes consistent positions throughout.

E.3 QUALITATIVE STREAMING EXAMPLES

We provide two visualizations of MindVLA-U1's streaming inference behavior to complement the architectural and ablation results.

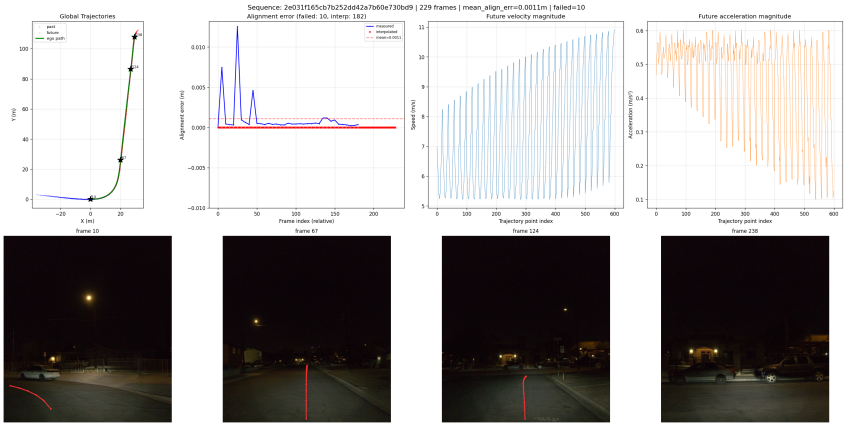


Figure 9: **Full-sequence pose recovery on a representative WOD-E2E segment** (229 **frames).** Top, left to right: recovered global trajectory in segment-anchor coordinates; per-frame SE(2) alignment residual (mean $\sim$0.0011 m, 10 inliers per join); speed-magnitude profile across the full sequence; acceleration-magnitude profile. Bottom: sampled front-view frames (#32, #67, #124, #198) with projected ego trajectory overlaid in red. Sub-cm alignment residual confirms that consecutive WOD-E2E clips can be stitched into a single global frame, enabling streaming training (§2.2) over full segments rather than isolated clips.

Per-frame streaming inference (Figure 10). Six consecutive frames from a single streaming sample are shown side-by-side, each column reporting one streamed frame: top — the front-view RGB input image; middle — the predicted BEV trajectory (5 s horizon, 20 waypoints) under the streaming forward pass; bottom (heat-mapped panels) — per-waypoint confidence over the prediction horizon. Predictions evolve smoothly across frames as the streaming memory channel propagates context, with no chunk-boundary discontinuities and no stale waypoints persisting across the stream.

Long-horizon trajectory consistency (Figure 11). Four full driving sequences (Seqs 33, 34, 35, 3) are each evaluated over 4 consecutive clips ($\sim$17 s, 68 predicted waypoints per sequence). For each sequence, the top row shows the per-clip predictions in their own local ego frames (Clip 0–3, each starting at the local origin and reset at clip boundaries); the middle panel stitches the four predictions into a single global coordinate system using the streaming pose chain; the bottom overlay compares the stitched prediction (per-clip colors) against the logged ground truth (green), reporting sequence-level ADE and FDE. The global-frame stitches stay coherent across clip boundaries despite per-clip ego-frame resets — sub-meter ADEs hold across all four sequences (right turn, leftward curve, curved cruise, sharp curving maneuver), and end-point errors stay within a few meters at the $\sim$17 s horizon.

F Mixture-of-Transformers Backbone

F.1 Architectural Details

This section gives the MoT routing details deferred from §2.1. All numbers below describe the deployed *(V,L)+(M,S,A)* grouping (the "Ours" row in Table 4).

Modality-routed attention. In the MoT variant, each layer splits attention into two independent groups: a *context* group serving visual and language tokens, and an *action* group serving memory, ego-state, and action tokens. The groups use independent Q/K/V/O projections, and the action group uses fewer attention heads (4) than the context group, which inherits the dense backbone's head count. Self-attention remains shared across the two groups within each layer, so the context representation is built with action context, preserving the unified-backbone property of §2.1.

Per-modality experts. The feed-forward stage replaces the single SwiGLU with group-specific experts. The context expert is cloned from the dense backbone and keeps its original intermediate width, while the action expert is initialized from scratch with a narrower intermediate width (d_{ff}=1024). Capacity is kept high for perception/language representation and made compact for motor decoding.

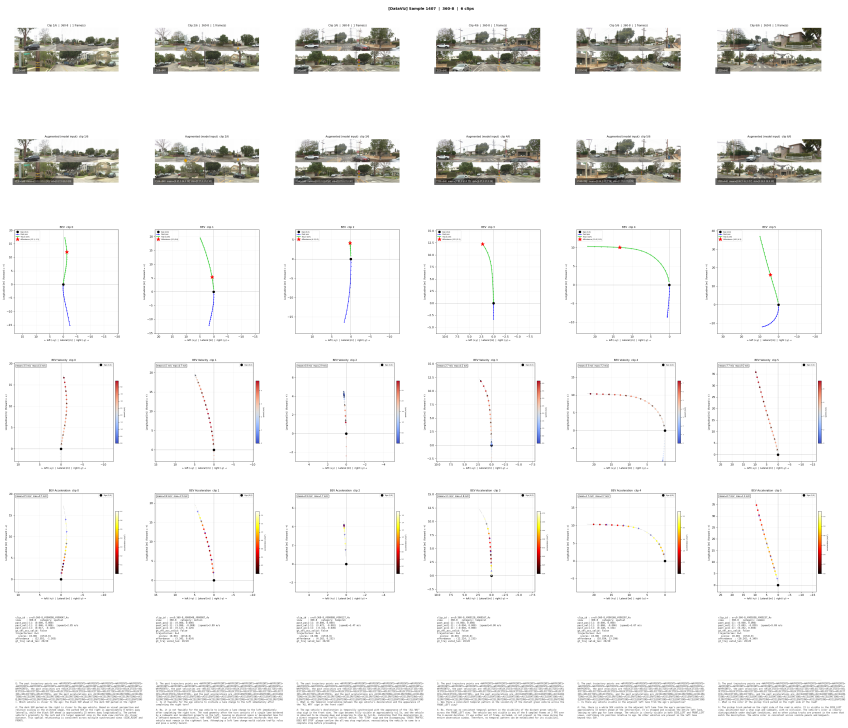


Figure 10: **Per-frame streaming inference across six consecutive frames of one streaming sample.** Per column: front-view input (top two rows), predicted BEV trajectory (middle), per-waypoint confidence heatmaps (bottom two rows). The streaming memory channel (§2.2, §E.1) carries scene context across frames; planned trajectories evolve smoothly with no fixed-chunk discontinuities.

Fast-mode subgraph. In fast mode, answer/thinking tokens are physically removed before the backbone pass. The remaining memory, ego-state, and action tokens route through the action group, while visual and question tokens are kept as conditioning prefix on the context side rather than decoded into language. This is different from mask-only fast/slow variants: the sequence length seen by attention is smaller, so the fast path can translate into actual compute reduction. The independence of the groups also makes temporal-frequency decoupling — caching context key-value states from a slow step and reusing them across subsequent fast steps — structurally straightforward; we leave this extension to future work. The throughput consequences of the dense vs. MoT fast paths are measured in §3.4 (Table 5).

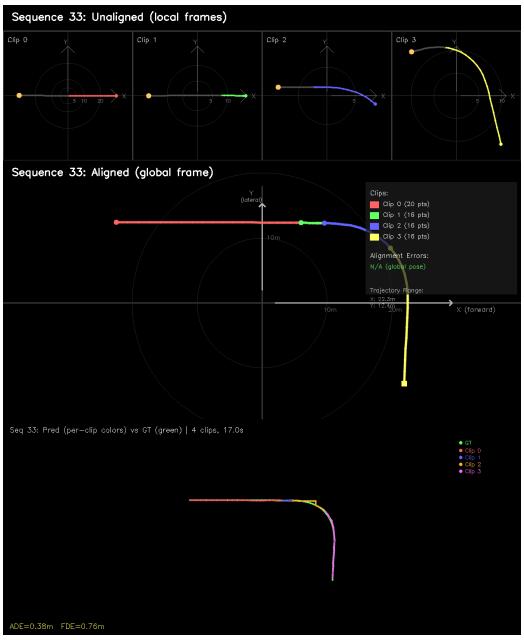
F.2 Foundation Architecture: Extension Beyond Driving

The deployed two-group MoT generalizes without changing the shared backbone: extensions enter as token-roles on the same K/V pool, not as new graph topologies. Figure 12 sketches the three-stage design space — perception (modality encoders), cognitive (context-group experts: language, reasoning, world model, safety), and action (action-group experts: behaviors, embodiments) — of which MindVLA-U1 populates one instantiation per stage (Camera; Language; Fast VA / Rich VLA). The other multiple instantiations are left to future work (§I).

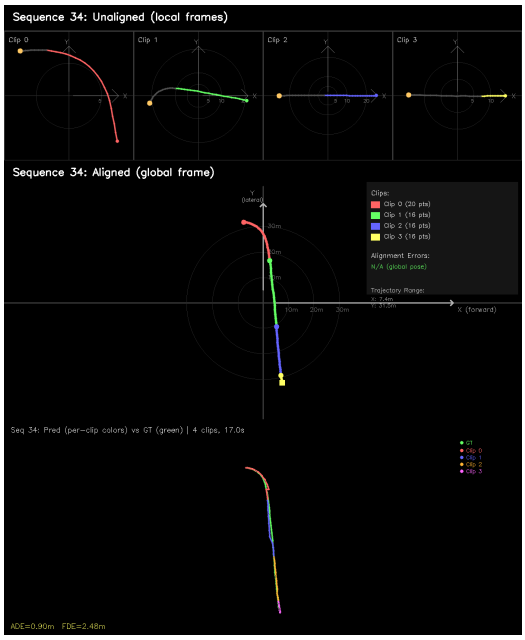
G RL Post-Training: Extended Details

This section gives the optimization hyperparameters, training-dynamics, checkpoint-selection, and ADE-rater trade-off details deferred from §3.6.

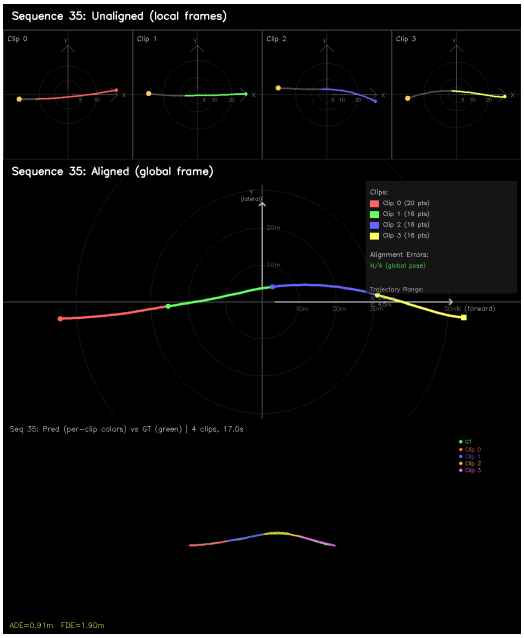
Hyperparameters. Optimizer: AdamW, learning rate 5×10^{-7} constant, β_1=0.9, β_2=0.999, weight decay 0.1, gradient norm clip 0.3. KL regulariser: β_{kl}=0.008 with the k3 estimator $\beta \cdot (\exp(r-c)-(r-c)-1)$ against a snapshot of the SFT weights. PPO clip $\epsilon = \pm$ 0.2. Rollouts: 8 trajectories per sample, group-scaled rewards. The RFS reward is computed at 4 Hz over a 5-second horizon. RL is run on 8 GPUs at batch size 1/GPU; RFS plateaus smoothly without the "collapse-then-



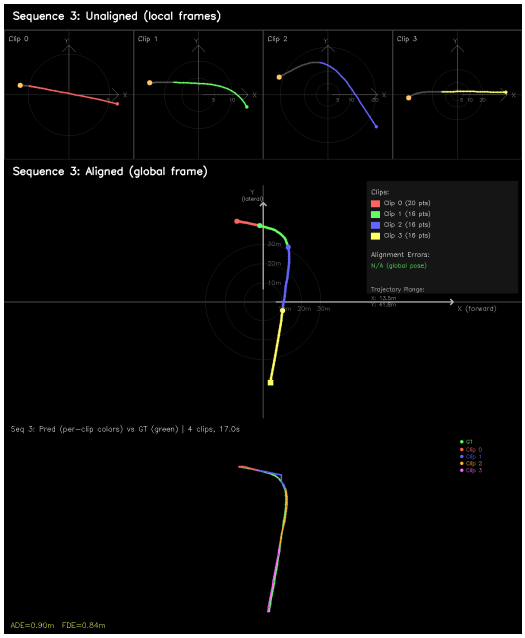
(a) Seq 33 — right turn (ADE 0.38 m, FDE 0.76 m)



(b) Seq 34 — leftward curve (ADE 0.90 m, FDE 2.48 m)



(c) Seq 35 — curved cruise (ADE 0.91 m, FDE 1.90 m)



(d) Seq 3 — sharp curving maneuver (ADE 0.90 m, FDE 0.84 m)

Figure 11: **Long-horizon streaming consistency over** 4 **consecutive clips (**∼17 **s,** 68 **waypoints).** Per sequence: top row shows per-clip predictions in their own local ego frames (Clips 0–3); middle stitches the four predictions in a single global frame via the streaming pose chain; bottom overlays the stitched prediction against the logged GT (green). Sub-meter ADEs hold across all four scenarios — right turn (a), leftward curve (b), curved cruise (c), and sharp curving maneuver (d) — with end-point errors remaining within a few meters at the ∼17 s horizon despite per-clip ego-frame resets at clip boundaries.

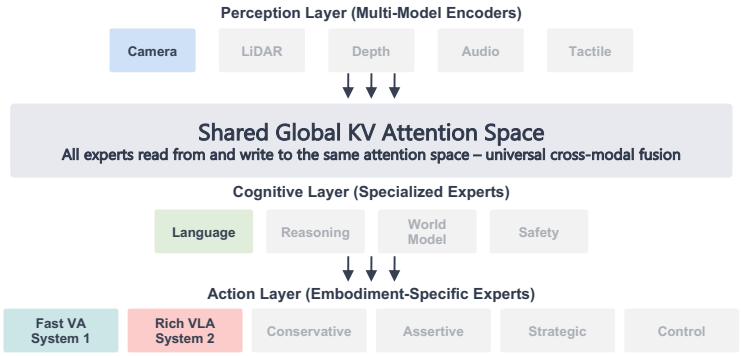


Figure 12: **Foundation architecture vision.** Three-stage generalization of the two-group MoT (§2.1, §F.1): perception, cognitive (context-group experts), and action (action-group experts). Highlighted: populated in MindVLA-U1; grey: extension slots on the same shared K/V pool.

recovery" pattern reported in some RL-from-rater systems, which we attribute to the conservative β_{kl} and the small constant learning rate.

ADE–rater trade-off under RFS-only reward. The detailed eval reveals an interesting trade-off that the headline RFS hides. RL post-training *improves* the rater-matched distance (minADE-5s drops $1.16 \rightarrow 1.07$, -0.09 m) while modestly *worsening* the GT-matched distance (ADE-5s rises $2.18 \rightarrow 2.22$, $+0.04$ m). The model has learned to plan trajectories closer to the rater panel, which is exactly what RFS rewards, at the cost of modest divergence from the single logged GT trajectory. This is consistent with the structure of the RFS reward: the ground-truth logged trajectory is one valid behavior but rarely the rater-preferred one in a multi-rater panel, so optimizing RFS pulls the model toward the panel mode. The effect is larger at the longer horizon, which we read as RL primarily sharpening late-trajectory rater alignment.

H Limitations

We list the scope conditions of the results above, framed as what the paper does *not* claim, so follow-up work can address each axis cleanly. *(i) Open-loop evaluation only.* WOD-E2E is logged, not reactive; the RFS metric and the rater panel test open-loop trajectory quality, not closed-loop policy quality under the model's own action distribution. Surpassing the 8.13 GT RFS establishes open-loop superiority; closed-loop on-vehicle behavior is a separate evaluation channel that we have not measured. *(ii) Single benchmark.* All headline results are on WOD-E2E. Cross-benchmark transfer to nuScenes, NAVSIM, or on-vehicle deployment is not yet verified; the architectural arguments generalize, but the empirical specifics may not.

I Future Roadmap

The limitations in §H invite natural follow-up directions on top of MindVLA-U1's unified interface; below we list several worth exploring:

1. Deeper RL post-training. The current RL stage is a single SFT→GRPO step on the RFS scalar (§3.6); the unified backbone admits richer reward sources, including rater-panel ensembles, and longer-horizon credit assignment, which we plan to explore.

2. Intent diversity and language alignment. Intent-CFG currently uses a 3-class GT-supervised intent; MindLabel's 20-class intent vocabulary and free-form language conditioning are natural extensions toward closing the predicted-intent vs. GT-intent gap of §3.3.

3. Cross-benchmark verification. Headline results in this paper are confined to WOD-E2E (limitation (ii)); extending evaluation to additional open AD benchmarks — nuScenes, NAVSIM, and similar — together with on-vehicle deployment would test whether the architectural arguments transfer empirically as well as conceptually.

4. Closed-loop training with a learned world model. A learned world model routed through the existing backbone as another modality — rather than a separate subsystem or a hand-built reactive simulator — would serve as the reactive environment in which the policy rolls out counterfactual futures under reward, addressing limitation (i) above.

5. MoT expansion to additional experts. The routing principle in §3.4 (perceptual modalities → context, motor/proprioceptive → action) extends naturally to additional context-group experts (*e.g.*reasoning, world-model rollout, safety) that share the K/V pool with the language expert but maintain independent FFN capacity; how far this extends in practice is an open question.

6. VLA inference acceleration. Several orthogonal speedups remain open for the fast path: distilling the 2-step flow into a single-step model, reusing the prefix KV cache across Euler steps and across consecutive frames, and decoupling the slow language pathway's update rate from the fast action loop. Each tightens the deployment budget without changing the unified architecture, and brings real-time control closer to VA-class throughput at the deployed VLA scale.

7. Transfer to embodied tasks. The unified-interface philosophy generalizes beyond driving; testing the same interface on manipulation and navigation is an important direction for future work.

The unified interface presented here is one step on the longer path toward fully autonomous driving (*i.e.*, L4); many more will be needed — in benchmarks, in on-vehicle validation, in closed-loop training with learned world models, and in directions we cannot yet anticipate. We present MindVLA-U1 in that spirit, and look forward to the perspectives, datasets, and use cases the community will bring — both to driving, and to the broader physical-intelligence agenda it ultimately serves.