Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail

NVIDIA[1]

Abstract

End-to-end architectures trained via imitation learning have advanced autonomous driving by scaling model size and data, yet performance remains brittle in safety-critical long-tail scenarios where supervision is sparse and causal understanding is limited. To address this, we introduce Alpamayo-R1 (AR1), a vision–language–action model (VLA) that integrates Chain of Causation reasoning with trajectory planning to enhance decision-making in complex driving scenarios. Our approach is built upon three key innovations: (1) the Chain of Causation (CoC) dataset, constructed through a hybrid auto-labeling and human-in-the-loop pipeline that produces decision-grounded, causally linked reasoning traces aligned with driving behaviors; (2) a modular VLA architecture combining Cosmos-Reason, a Vision Language Model (VLM) specifically pre-trained for Physical AI applications, with a diffusion-based action-expert trajectory decoder that generates kinematically feasible trajectories in real-time; (3) a multi-stage training strategy that first elicits reasoning capabilities through supervised fine-tuning on CoC data, then performs reinforcement learning (RL)-based post-training that optimizes reasoning quality via large reasoning model feedback and enforces reasoning-action consistency. Comprehensive evaluation demonstrates that AR1 achieves up to 12% improvements in trajectory planning on challenging cases, with a 35% reduction in off-road rate and 25% reduction in close encounter rate in closed-loop simulation. Furthermore, RL post-training improves reasoning quality by 45% while increasing reasoning-action consistency by 37%. Model scaling from 0.5B to 7B parameters shows consistent improvements, with the 7B model producing 11% more accurate trajectories compared to the 0.5B model. On-vehicle road tests confirm real-time performance (99ms end-to-end latency) and successful deployment in urban driving scenarios. By bridging interpretable reasoning with precise control, AR1 demonstrates a practical path toward Level 4 autonomous driving. We plan to release AR1 models and a subset of the CoC dataset soon.

1. Introduction

The evolution of autonomous driving systems has witnessed a paradigm shift from traditional modular architectures (Urmson et al., 2008; Paden et al., 2016; Zhang et al., 2018; Lefèvre et al., 2014) to end-to-end (E2E) driving frameworks (Bojarski et al., 2016; Hu et al., 2023; Gu et al., 2023; Weng et al., 2024; Wu, 2025), a transition increasingly embraced by industry. In contrast to modular designs that explicitly separate perception, prediction, and planning, E2E approaches employ a single neural network to map raw sensor inputs directly to control outputs. This unified formulation eliminates manually engineered interfaces, enabling joint optimization and data-driven policy learning at scale. Recent advances in transformer-based architectures, coupled with large-scale driving datasets have further improved the overall performance and generalization of the E2E driving paradigm. Despite these successes, current E2E approaches remain brittle in decision-making for long-tail and safety-critical scenarios, where supervision is sparse and high-level reasoning becomes essential. Consequently, a significant gap persists between the capabilities of existing E2E models and the requirements for achieving robust Level-4 autonomy with driving-specific reasoning capabilities.

Recent advances in large language models (LLMs) (Achiam et al., 2023; Comanici et al., 2025) offer a

[1]A detailed list of contributors and acknowledgments can be found in App. A of this paper.

promising direction to address this reasoning gap. LLMs have transformed artificial intelligence, with scaling laws (Kaplan et al., 2020) demonstrating that model performance improves predictably as compute and data increase. Beyond training-time scaling, recent frontier models such as OpenAI's o1 (OpenAI, 2024), DeepSeek-R1 (DeepSeek-AI, 2025), and similar systems have introduced a new paradigm: *inference-time reasoning*. Unlike traditional single-step answer generation, these models generate intermediate reasoning traces, denoted *chains of thought* (Wei et al., 2022), that mimic human problem-solving strategies. This shift makes inference time a tunable resource: allocating more compute to deliberative reasoning often yields more accurate, robust, and verifiable decisions (Yao et al., 2023). This reasoning capability is particularly important for autonomous driving, where decision-making is inherently uncertain and safety-critical. Text-based reasoning further enables models to explore alternative outcomes in language space before committing to actions, offering several key advantages:

(1) *improved safety* through explicit counterfactual reasoning and the potential for runtime safety cross-checks and monitoring;
(2) *better interpretability* via human-readable decision rationales;
(3) *richer training signals* that can be used as verifiable rewards to boost long-tail performance.

There exists recent work that integrates vision-language models (VLMs) and vision-language-action models (VLAs) into autonomous driving (Mao et al., 2023, 2024; Hwang et al., 2024; Zhou et al., 2025; Renz et al., 2025), however, most approaches either lack explicit reasoning (Wu, 2025; Zhou et al., 2025; Jiang et al., 2025) or perform reasoning in a free-form, unstructured manner (Luo et al., 2025; Yuan et al., 2025; Rowe et al., 2025). Such approaches struggle to generalize beyond training distributions, especially in ambiguous or compositional long-tail scenarios where strong domain priors are essential. Moreover, treating autonomous vehicle (AV) reasoning as a pure natural language processing (NLP) problem overlooks the rich structural knowledge inherent to driving: lane geometry, traffic rules, map priors, agent interactions, and kinematic constraints.

We argue that effective reasoning for autonomous driving must be *causally grounded* and *structurally aligned* with the task of driving. Instead of generating verbose, unstructured narratives, reasoning traces should explicitly link observed scene evidence to concrete driving decisions through causal chains, and these decisions should directly condition or control low-level trajectory generation. This design ensures that reasoning is not only an interpretability-enhancing addition, but a functional component that improves both training efficiency and closed-loop driving performance, particularly in safety-critical long-tail events.

In this work, we introduce **Alpamayo-R1**, a VLA that extends the vision-action (VA) model Alpamayo-VA (Wu, 2025) with structured reasoning capabilities, bridging reasoning and action prediction for generalizable autonomous driving. This model addresses the challenges stated above through three key innovations:

1. We develop a structured **Chain of Causation (CoC)** labeling framework that produces decision-grounded, causally-linked reasoning traces aligned with driving scenarios, supported by a hybrid human-in-the-loop and auto-labeling pipeline for scalable high-quality data generation.
2. We employ a **diffusion-based action-expert trajectory decoder** based on flow matching (Lipman et al., 2023; Driess et al., 2025) to efficiently generate diverse, continuous multi-modal trajectory predictions while maintaining alignment with the language-based reasoning outputs.
3. We perform **reinforcement learning (RL) post-training** to simultaneously improve reasoning quality and enforce consistency between textual reasoning and executed actions, using a composite reward that evaluates causal correctness, reasoning-action agreement, and trajectory safety.

Through extensive evaluation with open-loop metrics, closed-loop simulation, and on-vehicle tests, we demonstrate that AR1 achieves substantial improvements over end-to-end baselines, with the largest gains in rare, safety-critical scenarios.

In the following sections, we present the detailed components of our framework. Sec. 2 reviews related work. Sec. 3 introduces the model architecture with an analysis of various design choices. Sec. 4 describes the proposed hybrid labeling pipeline and the resulting CoC dataset, specifically developed for reasoning-based VLA tasks in autonomous driving. Sec. 5 outlines our multi-stage training strategy, where each stage progressively enhances the model's capabilities from improving general visual-language understanding in the AV domain, to generating action modalities, to strengthening reasoning ability and output alignment. Finally, Sec. 6 reports extensive evaluation results, demonstrating the effectiveness of our approach in both open-loop and closed-loop environments.

2. Related Work

Our work builds upon recent advances in VLMs for autonomous driving, reasoning-augmented action models, and post-training alignment techniques. We organize our review around four key areas. First, we discuss the evolution from general-purpose VLMs (Hwang et al., 2024; Xu et al., 2024) to action-oriented VLAs (Zhou et al., 2025; Renz et al., 2025) in autonomous driving (Sec. 2.1), highlighting the shift toward embodied action prediction. Second, we examine reasoning VLAs (Sec. 2.2) that incorporate explicit chain-of-thought processes (Wei et al., 2022; Luo et al., 2025; Rowe et al., 2025) for interpretable decision-making. Third, we review post-training alignment methods (Sec. 2.3), particularly RL from human feedback (RLHF) (Christiano et al., 2017) and RL with verifiable rewards (RLVR) (DeepSeek-AI, 2025), which form the foundation of our reasoning alignment approach. Finally, we review vision-language datasets in autonomous driving (Sec. 2.4), identifying key limitations in existing reasoning datasets (Sima et al., 2024; Nie et al., 2024) that motivate our data construction methodology.

2.1. VLMs and VLAs in Autonomous Driving

Early work explored leveraging LLMs' general knowledge for driving. Drive-GPT (Mao et al., 2023), Wolf (Li et al., 2025), and AgentDriver (Mao et al., 2024) treat planning as text generation or language-based tool use, achieving competitive open-loop performance. Cube-LLM (Cho et al., 2025), TOKEN (Tian et al., 2024), and EMMA (Hwang et al., 2024) scale multimodal LLMs to multi-task scene understanding and trajectory prediction. VLM-AD (Xu et al., 2024) uses VLMs as training-time supervisors, while ReAL-AD (Lu et al., 2025) models hierarchical reasoning, and DiMA (Hegde et al., 2025) distills VLM knowledge into efficient, LLM-free planners.

A complementary line of work couples language with explicit action representation to create VLA models. OpenDriveVLA (Zhou et al., 2025) autoregressively produces trajectory waypoints from structured vision-language tokens. AutoVLA (Zhou et al., 2025) unifies reasoning and action with adaptive "think vs. act" control. IRL-VLA (Jiang et al., 2025) incorporates inverse RL for safety-efficiency balance, CoReVLA (Fang et al., 2025) targets long-tail scenarios, and SimLingo (Renz et al., 2025) achieves state-of-the-art closed-loop results in Bench2Drive (Jia et al., 2024). However, these approaches largely operate reactively without explicit reasoning, struggling to generalize beyond training distributions in ambiguous or long-horizon scenarios requiring counterfactual reasoning.

2.2. Reasoning VLAs in Autonomous Driving

Explicit reasoning methods such as chain-of-thought (CoT) (Wei et al., 2022) and tree-of-thought (ToT) (Yao et al., 2023) have demonstrated that intermediate reasoning traces can substantially improve performance in complex language tasks. In the domain of autonomous driving, many recent works on VLA adopt this insight by integrating structured reasoning into vision-to-action pipelines. One line of work focuses on adaptive or efficient invocation of reasoning. For example, AdaThinkDrive (Luo et al., 2025) uses a fast-and-slow thinking mechanism, trained with RL, to invoke CoT only when needed, reducing inference overhead while maintaining performance. AutoDrive-R^2 (Yuan et al., 2025) builds an explicit CoT and self-reflection dataset (nuScenesR^2-

6K), leveraging GRPO (Shao et al., 2024) with physics-grounded rewards to refine reasoning-augmented trajectories while ensuring physical feasibility.

Other approaches explore diverse reasoning strategies: RIV-CoT (Corbière et al., 2025) augments CoT with retrieval, FutureSightDrive (Zeng et al., 2025) performs spatio-temporal reasoning, and CoT-Drive (Liao et al., 2025) distills reasoning into lightweight models. ReCogDrive (Li et al., 2025), ReasonPlan (Liu et al., 2025), MTRDrive (Luo et al., 2025), Drive-R1 (Li et al., 2025), AgentThink (Qian et al., 2025), DriveAgent (Hou et al., 2025), and DSDrive (Liu et al., 2025) combine memory, tool invocation, multi-agent reasoning, or compression. Notably, Poutine (Rowe et al., 2025) topped the 2025 Waymo Vision-Based End-to-End Driving Challenge, demonstrating that reasoning-enhanced VLAs with RL finetuning excel in long-tail scenarios. This work demonstrates that reasoning serves as a functional core of driving decisions, with trade-offs among interpretability, runtime cost, and performance. However, most existing approaches rely on free-form reasoning that lacks explicit causal grounding and consistency between reasoning and actions. In contrast, our work introduces a structured CoC framework that ties reasoning to concrete driving decisions, and employs post-training RL to simultaneously optimize reasoning quality, reasoning-action consistency, and trajectory safety.

2.3. Post-training Alignment

Generative models (e.g., LLMs and text-to-image generators) are predominantly trained with an imitative objective, such as next-token prediction. While this objective enables efficient learning from Internet-scale data, it remains only a proxy for the true training goal: optimizing for the expert's internal reward function that motivated the demonstrated behavior. Consequently, generative models may deviate from end-user intent and, in some cases, exhibit safety-critical failures, such as producing harmful text outputs (Mu et al., 2024), unsafe visual generations (Lee et al., 2023), or hazardous robot motions (Lu et al., 2023). To mitigate such misalignment, post-training alignment—particularly through Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017) has emerged as a central strategy for aligning generative models with human preferences. For reasoning models specifically, DeepSeek-R1 (DeepSeek-AI, 2025) employs Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to directly improve reasoning quality by rewarding verifiable solutions rather than intermediate token likelihood, while OpenAI o1 (OpenAI, 2024) similarly demonstrates that outcome-based RL substantially enhances chain-of-thought quality. In the embodied AI domain, these alignment techniques have been extended to VLAs to generate actions that better reflect human intent across diverse embodiments, including autonomous driving (Tian and Goel, 2025) and assistive robots (Tian et al., 2024; Zhang et al., 2025). While these methods focus on improving action outcomes, our work addresses a complementary dimension: improving the reasoning process itself and ensuring that the model's internal decision rationale remains causally consistent and contextually grounded in the context of safety-critical autonomous driving.

2.4. Vision-Language Datasets for Autonomous Driving

Building upon the open-source nuScenes (Caesar et al., 2020) dataset, early work (Qian et al., 2024; Wu et al., 2025; Tian et al., 2025) primarily focuses on object-centric perception tasks, enabling VLMs to acquire general perception knowledge and improve object grounding in driving scenes. Beyond nuScenes, datasets such as WOMD-reasoning (Li et al., 2024) and DriveQA (Wei et al., 2025) extend vision-language annotations to large-scale motion datasets such as the Waymo Open Motion Dataset (Ettinger et al., 2021) and the CARLA simulator (Dosovitskiy et al., 2017), focusing on describing interactions between agents, traffic rules, and right of way principles. While these datasets serve as valuable resources for VLM pre-training, their language annotations are not explicitly linked to the ego-vehicle's actions. As a result, they provide limited supervision for *planning-oriented* reasoning, a key capability required by VLAs. To bridge this gap, prior work has focused on constructing language datasets tailored for motion planning. For instance, Drama (Malla et al., 2023) annotates important objects that may influence the ego vehicle's behavior. Subsequent works such as DriveAction (Hao et al., 2025) and DriveBench (Xie et al., 2025) develop comprehensive QA pairs for VLA training, emphasizing

not only the identification of critical objects for planning, but also covering QA pairs for motion prediction, traffic signs, road markings, navigation following, etc.

Motivated by the development of reasoning VLAs, recent research has shifted from general VLA datasets to reasoning-oriented ones, where explicit explanations are provided for the ego vehicle's actions. As an early effort, BDD-X (Kim et al., 2018) provides a small set of human-written explanations describing driver behaviors. With the significant advancement of LLMs/VLMs, subsequent works such as DriveGPT4 (Xu et al., 2024), CoVLA (Arai et al., 2025), and LingoQA (Marcu et al., 2024) introduce automated or human-in-the-loop pipelines to enrich the linguistic expressiveness of reasoning data. To capture the full reasoning process across perception, prediction and planning, DriveCoT (Wang et al., 2024), Nuinstruct (Ding et al., 2024), Reason2drive (Nie et al., 2024), DriveLM (Sima et al., 2024), Drivelmm-o1 (Ishaq et al., 2025), and Senna (Jiang et al., 2024) develop explicit chain-based reasoning pipelines for data construction. In parallel, Impromptu VLA (Chi et al., 2025) focuses on curating reasoning data in unstructured road scenarios. However, these datasets still exhibit key limitations in enforcing the causal relationship between observations and actions in their reasoning traces. For example, free-form reasoning traces tend to use vague descriptions such as *"the ego vehicle should be cautious and watch out for ..."* rather than specifying actionable driving decisions. Additionally, many reasoning traces contain superficial causal factors such as *"sunny weather", "wide roads", "... due to traffic rules"*, or introduce causal confusion by exposing the entire video clip in the labeling process and referencing future events that are not observable. These issues underscore the need for a dataset with explicit, decision-grounded, and causally linked reasoning traces, motivating our proposed CoC data pipeline.

3. Building a Reasoning VLA Architecture

Building an effective and reasoning-capable VLA for autonomous driving requires enabling several new capabilities beyond what general-purpose VLMs (Achiam et al., 2023; Comanici et al., 2025) currently offer. First, autonomous vehicles rely on *multi-camera, multi-timestep observations* to achieve 360-degree situational awareness, yet standard VLMs typically process images or video frames *independently* without explicit temporal or cross-view reasoning, leading to prohibitive token counts that preclude real-time inference when handling multi-camera inputs. Second, driving decisions must be grounded in *causally structured reasoning* (Wei et al., 2022) rather than free-form narratives; the model must explain *why* a maneuver is safe and legal based on observable evidence in the history window. Third, the model must generate *precise, multi-modal trajectory predictions* in real-time; autoregressively decoding waypoints as text tokens is inefficient and lacks the geometric and kinematic constraints essential for safe vehicle control (Driess et al., 2025). Finally, to ensure safety in long-tail scenarios, reasoning traces must be *aligned with executed actions*.

To address these challenges, we introduce **Alpamayo-R1 (AR1)**, a modular VLA architecture that extends Alpamayo-VA (Wu, 2025) to integrate reasoning with action prediction for autonomous driving. Our design philosophy emphasizes *flexibility* and *modularity*: the architecture can adopt any off-the-shelf VLM backbone while incorporating domain-specific components for efficient vision encoding and real-time action decoding. This modularity enables us to leverage advances in vision-language pretraining (NVIDIA et al., 2025; Bai et al., 2025) while efficiently bridging high-level reasoning with low-level control for autonomous driving.

Problem Formulation. Given a sequence of past observations $\boldsymbol{o}$ up to timestamp T (omitted below), including multi-camera images $\boldsymbol{o}_{\text{image}}$ and egomotion history $\boldsymbol{o}_{\text{egomotion}}$, AR1 is trained to perform reasoning, denoted as $\textsc{Reason}$, and to predict the future trajectory of the ego vehicle $\boldsymbol{\tau}$. We formulate this task as a sequential prediction problem, where the entire sequence is constructed as

$$[\boldsymbol{o}_{\text{image}}, \boldsymbol{o}_{\text{egomotion}}, \textsc{Reason}, \boldsymbol{\tau}], \tag{1}$$

with each component conditioned on all previous ones. By default, the model is trained to predict the entire

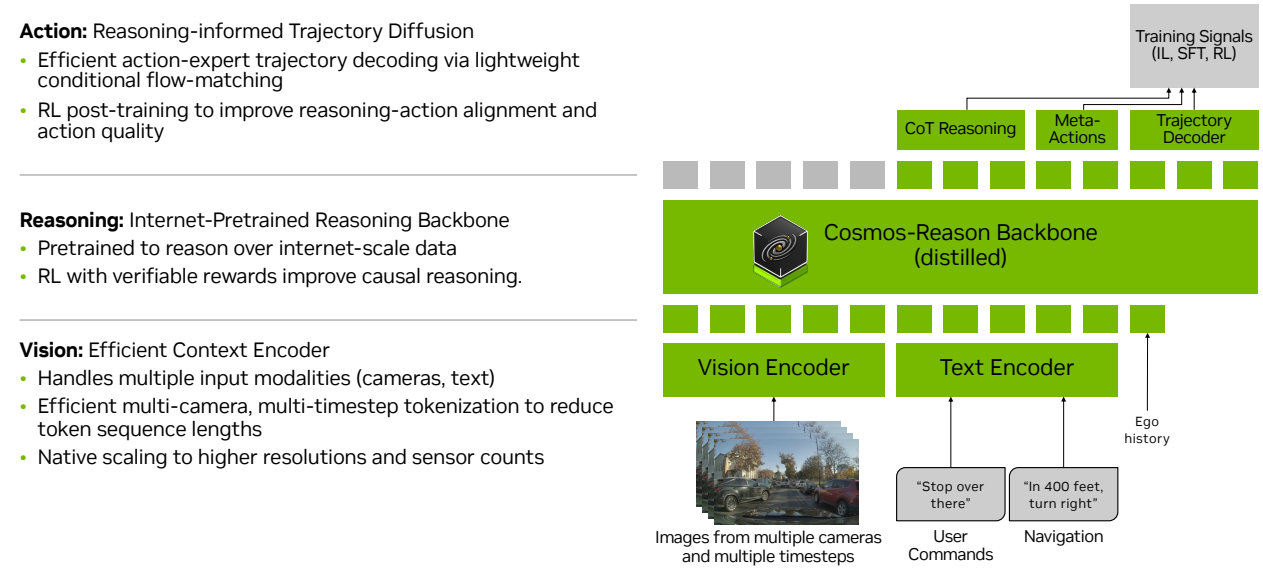

Figure 1: Overview of Alpamayo-R1 architecture. Multi-camera images and egomotion are processed by a vision encoder to produce visual tokens, which are fed into a VLM backbone (Cosmos-Reason) along with textual inputs. The model autoregressively generates chain-of-thought reasoning and discrete trajectory tokens. At inference, an action-expert decoder using flow matching converts the discrete trajectory tokens into continuous, kinematically feasible waypoints conditioned on the reasoning output.

6s-long future trajectory sequence

$$\boldsymbol{\tau} = \{(x^i, y^i, \theta^i_{\text{yaw}})\}_{i=1}^{64}, \tag{2}$$

where $(x^i, y^i, \theta^i_{\text{yaw}})$ denotes the i-th future waypoint sampled at 10 Hz in the ego-vehicle's coordinate frame at time T. As will be detailed in Sec. 3.2.2, we adopt a control-based representation using unicycle dynamics with control inputs

$$\boldsymbol{a} = \{(a^i, \kappa^i)\}_{i=1}^{64}, \tag{3}$$

where a^i and κ^i denote the acceleration and curvature at timestep i, respectively. Details of how $\boldsymbol{\tau}$ is encoded and decoded are provided in Sec. 3.2.2 and 5.1.

Overall Architecture. Fig. 1 presents the end-to-end architecture of AR1. The system processes multi-camera, multi-timestep observations as visual inputs, optionally augmented with textual inputs such as user commands and high-level navigation instructions. All inputs, including historical ego-motion data, are tokenized into a unified sequence of multimodal tokens following a predefined order. These tokens are then processed by the Cosmos-Reason (NVIDIA et al., 2025) backbone, which produces output tokens representing reasoning traces, meta-actions, and predicted future trajectories. The model is trained in multiple stages, combining supervised fine-tuning (SFT) and RL, as will be described in Sec. 5.

3.1. VLM Backbone: Cosmos-Reason

We adopt Cosmos-Reason (NVIDIA et al., 2025) as the VLM backbone for Alpamayo-R1. Cosmos-Reason is a VLM specifically designed for Physical AI applications, post-trained on 3.7M Visual Question Answering (VQA) samples to develop physical common sense and embodied reasoning capabilities. The model incorporates 24.7K curated video VQA samples focused on driving scenarios, including scene descriptions, driving difficulty annotations, and reasoning traces distilled from DeepSeek-R1 (DeepSeek-AI, 2025) to predict the next action.

Domain-Specific Supervised Fine-Tuning. To further enhance Cosmos-Reason for autonomous driving deployment, we curate supplementary datasets spanning multiple Physical AI domains, including autonomous driving, robotics, healthcare, smart cities, manufacturing, retail, and logistics. This broad Physical AI pre-

training enables the model to develop general physical common sense and embodied reasoning capabilities that transfer to driving scenarios. For autonomous driving specifically, we augment the training data with 100K new samples that include annotations for critical objects in the environment and reasoning for the next action.

Driving-Focused Data Curation. We develop complementary labeling approaches to balance quality and scale for autonomous driving:

- *Human-labeled data* includes comprehensive annotations covering the operational design domain (weather, lighting, road conditions), traffic regulations (traffic lights, signs), ego behaviors (interactive and non-interactive meta-actions), critical objects influencing ego behavior, and causal reasoning behind observed maneuvers. These labels improve the model's understanding and reasoning in complex driving scenarios.
- *Automatically labeled data* focuses on ego behavior reasoning and prediction, generated by prompting a teacher VLM (e.g., Qwen3-VL (Qwen Team, 2025)) with driving-specific priors that encode longitudinal, lateral, and lane-related meta-actions along with velocity information. This scalable approach strengthens the model's predictive reasoning capabilities.

3.2. Domain-Specific Adaptations

While Cosmos-Reason provides a strong foundation, two critical gaps remain for real-world autonomous driving deployment: efficient vision encoding for multi-camera, multi-timestep inputs and precise trajectory decoding for real-time control. The following subsections detail our domain-specific components that address these challenges.

3.2.1. Vision Encoding

The main role of vision encoders within VLMs is to convert input images into streams of tokens for later processing by the LLM backbone. However, as VLAs target onboard deployment, a critical requirement of their vision encoders is to produce as few tokens as possible while preserving relevant semantic information from the environment. To achieve this, there have been a variety of vision tokenization approaches proposed that primarily differ in how much information is encoded per inference step (i.e., how many images are compressed into how many tokens), as well as their associated architectural choices.

In this section, we discuss the different vision encoders that AR1 can use as well as their tradeoffs, in addition to avenues for further token count compression towards enabling real-time onboard inference with larger backbone sizes.

Single-Image Tokenization. Many vision tokenizers primarily focus on representing single images and either employ autoencoding architectures (Sohn et al., 2015; van den Oord et al., 2017; Esser et al., 2021) or directly encode patches of pixels (Dosovitskiy et al., 2020). VLMs adopt the latter primarily and employ Vision Transformers (ViTs) (Dosovitskiy et al., 2020) to partition images into patches that are encoded to form a 1D token sequence. We denote this paradigm as *single-image tokenization*, where a model encodes each input frame into a set of tokens.

AR1's default tokenizer (and the one used for all subsequent experiments) leverages this paradigm, employing the base VLM's vision encoder (e.g., Zhai et al. (2023); Tschannen et al. (2025)) to encode a $W \times H$ px input image into patch features $\mathbf{f} \in \mathbb{R}^{W/14 \times H/14 \times D}$ which are then $2\times$ bilinearly downsampled to $\mathbf{f}' \in \mathbb{R}^{W/28 \times H/28 \times D}$ features per image. As an example, with $W = 448, H = 280$ this process produces 160 tokens per image.

Multi-Camera Tokenization. While single-image tokenization is simple to implement, it produces token counts that scale linearly with image resolution and the number of cameras (Wang et al., 2025). To obtain a 360-degree view of their surroundings, AVs often use 6 to 10 cameras, the patch-based tokenization of which would yield thousands of tokens per timestep and preclude real-time inference. Accordingly, AR1 also supports the use of a new line of efficient multi-camera tokenizers that encode images from multiple cameras into an intermediate representation before tokenizing that representation.

Specifically, AR1 can additionally use the efficient multi-camera tokenizer proposed in Ivanovic et al. (2025), which leverages triplanes as a 3D inductive bias, to simultaneously represent multiple camera images in an efficient manner. Crucially, since the triplane sizes are fixed, the input number of cameras and their resolution are decoupled from the resulting number of tokens. Formally, for a triplane with grid sizes S_x, S_y, S_z and downstream patchification values of p_x, p_y, p_z, the number of tokens produced by the tokenizer is

$$\underbrace{\left(\frac{S_x - p_x}{p_x} + 1\right)\left(\frac{S_y - p_y}{p_y} + 1\right)}_{\#\text{ of patches in the } xy \text{ plane}} + \underbrace{\left(\frac{S_x - p_x}{p_x} + 1\right)\left(\frac{S_z - p_z}{p_z} + 1\right)}_{\#\text{ of patches in the } xz \text{ plane}} + \underbrace{\left(\frac{S_y - p_y}{p_y} + 1\right)\left(\frac{S_z - p_z}{p_z} + 1\right)}_{\#\text{ of patches in the } yz \text{ plane}}. \quad (4)$$

As an example, for $S_x = S_y = 96, S_z = 48$, and $p_x = p_y = p_z = 8$, only 288 tokens are needed to represent one timestep of observations, irrespective of the number of cameras or their resolution. For a 7-camera vehicle setup, this equates to approximately 41.1 tokens per image ($3.9\times$ less than single-image tokenization). Further, as we will show in Sec. 6.6, this efficiency is achieved without major compromises to end-to-end driving metrics.

Multi-Camera Video Tokenization. While the above already yields significant reductions in the number of tokens required to represent sensor observations, there are still two fundamental areas where additional efficiency can be achieved:

(1) accounting for temporal information (e.g., there can be redundancy in information across frames);
(2) removing the potential performance ceiling that comes with using a structured feature representation.

Accordingly, AR1 also supports using multi-camera video tokenizers that directly encode entire sequences of camera observations from multiple timesteps. One example is Flex (Yang et al., 2025), which compresses a set of image tokens from multiple cameras and timesteps via full self-attention layers and a fixed set of query vectors, providing an explicit mechanism to control the magnitude of the information bottleneck. As will be shown in Sec. 6.6, this approach can achieve an up to $20\times$ token compression rate (compared to single-image tokenization) while maintaining or even improving downstream driving metrics.

Additional Avenues for Token Compression. Beyond the tokenization strategies described above, several complementary approaches can further reduce token counts. Post-training token pruning techniques, exemplified by SparseVILA (Khaki et al., 2025), dynamically identify and remove redundant tokens during inference without retraining, offering a practical path to reduce computational costs on models already trained. These methods represent promising directions for further scaling AR1 to even larger backbones while maintaining real-time performance constraints.

3.2.2. Trajectory Decoding

To extend the capability of a VLM to operate effectively in the physical world, it is essential to incorporate *physical actions*, corresponding to future driving trajectories in the autonomous driving context, into the training of the VLA. However, embodiment introduces unique challenges in action decoding:

(1) the action representation must be accurate, preserving both fidelity and multi-modality;
(2) the decoding process must be fast enough to support real-time inference;
(3) the decoding mechanism should integrate seamlessly into the VLA training pipeline.

Initially, we found that training the model in raw position (i.e., x, y) waypoint space is susceptible to sensor noise, which often degrades model convergence. Moreover, the downstream low-level vehicle controllers typically smooth trajectory outputs to ensure consistent and stable execution on-vehicle. Thus, instead of directly learning $\boldsymbol{\tau}$ in the raw position waypoint space, we adopt an action representation governed by unicycle dynamics that leads to better closed-loop performance. Specifically, we employ the following unicycle dynamics

with control input $\boldsymbol{a} = \{(a^i, \kappa^i)\}_{i=1}^{64}$ (Lynch and Park, 2017) and apply Euler discretization:

$$\mathbf{x}^{i+1} = \begin{pmatrix} x^{i+1} \\ y^{i+1} \\ \theta^{i+1} \\ v^{i+1} \end{pmatrix} = \begin{pmatrix} x^i + \frac{\Delta T}{2}\left(v^i \cos\theta^i + v^{i+1}\cos\theta^{i+1}\right) \\ y^i + \frac{\Delta T}{2}\left(v^i \sin\theta^i + v^{i+1}\sin\theta^{i+1}\right) \\ \theta^i + \Delta T\,\kappa^i v^i + \frac{\Delta T^2}{2}\kappa^i a^i \\ v^i + \Delta T\, a^i \end{pmatrix}, \tag{5}$$

where $\Delta T = 0.1$s in our setup, x and y denote positional waypoints in the bird's-eye-view (BEV) plane, θ represents the yaw angle, v the velocity, κ the curvature, and a the acceleration. During training, the ground-truth control sequence $\boldsymbol{a}$ is derived from $\boldsymbol{\tau}$ through a least-squares formulation with Tikhonov regularization to attenuate high-frequency noise. The model is trained to predict the control sequence $\boldsymbol{a}$ and, during inference, we apply Eq. (5) to map it to $\boldsymbol{\tau}$.

Furthermore, to enable AR1 to understand and generate trajectories, we encode $\boldsymbol{\tau}$ either as discrete tokens or continuous embeddings. In the discrete representation, we uniformly quantize each continuous value in $\boldsymbol{a}$ within a predefined range into equally spaced bins and represent the resulting indices as special tokens. For the continuous representation, we map $\boldsymbol{a}$ into AR1's embedding space using sinusoidal positional encoding followed by an MLP projection. Specifically, we adopt a strategy inspired by $\pi_{0.5}$-KI (Driess et al., 2025), combining *discrete* trajectory tokens learned within the VLM with an action-expert that decodes the same trajectories into *continuous* representations using a flow matching framework (Lipman et al., 2023). This framework facilitates streamlined VLM training, accelerates trajectory decoding, and achieves better closed-loop performance. Training details of the action modality injection are provided in Sec. 5.1.

Summary. This section further detailed the two principal design dimensions (vision encoding and action decoding) through which VLMs can be systematically adapted into AV policy VLAs. In subsequent sections, we detail the construction of the data pipeline and the formulation of the training strategy, which together endow the model with enhanced reasoning and alignment capabilities, thereby improving its robustness in handling long-tail events.

4. Chain of Causation Dataset: Learning Causally Grounded Reasoning VLAs

To enable reasoning VLA models to explain the causes of driving actions and to improve their trajectory-level performance, reasoning data must be closely correlated with the ego trajectory. However, existing CoT reasoning datasets in the AV community often exhibit several limitations, as shown in Fig. 2:

(1) *Vague behavior descriptions*: free-form CoT annotations may fail to specify concrete driving actions or may choose words that weakly correlate with ego trajectories;
(2) *Superficial reasoning*: some reasoning traces primarily describe contextual observations or hypothetical factors that lack a direct causal link to the ego vehicle's behavior, providing limited benefit for improving post-training driving performance;
(3) *Causal confusion*: reasoning traces may include causal factors that occur in future time windows, which are not observable to the model during training. This arises because the labeling process often exposes the entire video without distinguishing between historical and future segments.

To address these gaps, we introduce a labeling framework that enforces an explicit causal structure in the reasoning traces. We first define a comprehensive set of high-level driving decisions that directly correspond to low-level ego trajectories. Each reasoning trace is associated with an explicit driving decision and includes only the causal factors that motivate that driving decision. By carefully selecting keyframes to split historical and future video segments, we ensure that all causal factors originate within the observable history window,

Figure 2: Examples of reasoning traces exhibiting common issues in existing datasets (Malla et al., 2023; Chi et al., 2025; Arai et al., 2025). Text highlighted in yellow indicates vague behavior descriptions that fail to specify concrete driving decisions correlated with the trajectories. Text highlighted in blue denotes superficial reasoning, such as contextual observations that do not directly inform the ego vehicle's decision. Red highlights indicate incorrect or causally inconsistent reasoning that contradicts the actual behavior of the ego vehicle.

thereby preventing causal confusion. This design ensures that every reasoning trace is both *decision-grounded* and *causally linked*, capturing concise and interpretable cause–and–effect relationships rather than verbose descriptive narratives. The resulting dataset, termed the **Chain of Causation (CoC) dataset**, provides clear supervision for learning decision causality, enabling reasoning VLAs to efficiently reason about the causes of specific driving actions during onboard inference. An overview of our labeling pipeline is shown in Fig. 3.

4.1. Structured Chain of Causation

To facilitate efficient annotation, our labeling framework decomposes each data sample into three structured components: the driving decision, the causal factors (critical components), and the composed CoC trace. Consequently, each data instance constitutes a structured CoC sample encompassing these three components.

Driving Decision. To ensure our CoC data is *decision-grounded*, we define a closed set of high-level driving decisions as in Tab. 1. Each clip is annotated with at most one longitudinal and one lateral decision (or *None* for either channel), corresponding to the first action taken by the ego vehicle immediately after the critical reasoning moment. This standardized inventory directly aligns with low-level trajectories and eliminates free-form, vague descriptions of driving behavior, ensuring that every reasoning trace unambiguously specifies *what* decision is taken. For linguistic consistency and diversity, the final CoC reasoning traces are constructed using a compact verb set aligned with these driving decisions.

Critical Components. In contrast to the closed-set driving decisions, causal factors are defined as an open-ended set, with categories and example attributes described in Tab. 2. This design allows human labelers or an auto-labeling pipeline to flexibly specify only the key elements that directly influence the driving decision, while maintaining a structured output.

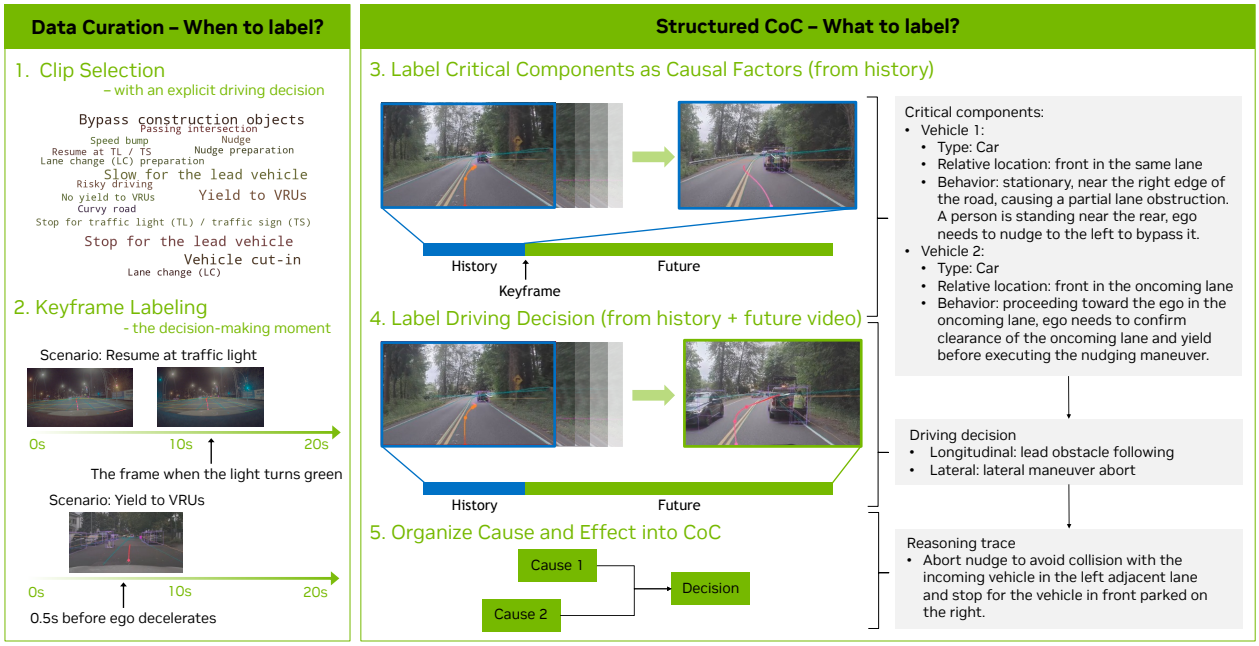

Figure 3: Overview of the proposed structured CoC labeling pipeline, composed of five steps: (1) **Clip Selection**, where clips containing explicit driving decisions are selected, filtering out low-signal clips that offer limited causal information; (2) **Keyframe Labeling**, where the decision-making moment within each video clip is identified, minimizing potential causal confusion; (3-5) **Structured CoC Labeling**, to construct the final CoC and further mitigate causal confusion, we first annotate critical components from the observation while avoiding referring to causal factors in future frames, and then label the corresponding driving decision. We then compose a reasoning trace from driving decisions and causal factors in natural language.

Composed CoC Traces. Once the driving decision and critical components are identified, they are linguistically organized into a coherent CoC reasoning trace that captures the causal rationale behind the chosen decision. As a result, the structured CoC protocol enforces:

(1) *decision grounding*: each reasoning trace is anchored to a single, explicit decision at the critical moment;
(2) *causal locality*: all evidence must originate from the observed history window;
(3) *annotation economy*: only decision-relevant factors are included.

4.2. Data Curation

Having defined the structured components of CoC (driving decisions, critical components, and composed CoC traces), the next step is to determine when these reasoning data should be labeled. Not every video clip warrants annotation; labeling is triggered only at moments where a clear causal link can be established between observable factors and the ego vehicle's subsequent decision. Therefore, a key aspect of our data labeling framework is data curation, which involves identifying these critical reasoning moments.

Clip Selection. We choose clips that contain an explicit driving decision to label the CoC dataset, thereby avoiding low-signal clips that provide limited causal information. These clips are categorized into two types of scenarios: (1) *Reactive* - where the ego vehicle must immediately adapt its behavior in response to a specific event, such as stopping for a lead vehicle or red light, or adjusting its lateral position to maintain clearance from a nearby obstacle or hazard; (2) *Proactive* - where the ego vehicle is not required to react instantly but must actively assess and anticipate potential maneuver adjustments due to upcoming road events or obstacles. For example, the ego may receive a routing command to change lanes but lacks sufficient space in the target lane, requiring continuous gap searching and space assessment in preparation for the lane change maneuver. We employ rule-based methods to identify clips corresponding to each scenario and balance the number of

Type	Driving decision	Definition
Longitudinal	Set speed tracking	Maintain or reach a target speed when unconstrained; excludes follow/yield/stop logic.
	Lead obstacle following	Maintain a safe time gap to the lead entity (closest in-path entity moves in the same traffic flow); excludes geometry-based slowing, gap-matching, and yielding to non-lead entity.
	Speed adaptation (road events)	Adjust speed for roadway features (curves, grades, bumps, ramps, roundabouts, turns); independent of a lead.
	Gap-searching (for LC/merge/zipper)	Adjust speed to match the target stream or create a usable gap to support a planned lateral maneuver.
	Acceleration for passing/overtaking	Increase speed to pass a slower lead with an associated lateral plan.
	Yield (agent right-of-way)	Slow/stop to concede priority to specific agents (pedestrians, cross-traffic, emergency vehicles, cut-ins).
	Stop for static constraints	Decelerate to—and hold at—control points (stop/yield lines, red light, school bus/rail rules); Sometimes a yield is necessary even when owning the right-of-way, to avoid a collision.
Lateral	Lane keeping & centering	Maintain position within lane boundaries; minor in-lane offsets allowed; never cross lane lines.
	Merge / Split (facility change)	Transition between facilities (e.g., on-ramp $\leftrightarrow$ mainline, weave segments); not a same-road lane change.
	Out-of-lane nudge (straddle avoidance)	Brief, intentional lane-line crossing to increase clearance around a blockage/hazard; return to original lane; specify left/right.
	In-lane nudge	Temporary offset within the lane (no line crossing) to increase clearance around a blockage/hazard; specify left-/right.
	Lane change (lateral push)	Full adjacent-lane transition with gap negotiation; specify left/right in reasoning trace.
	Pull-over / curb approach	Move toward edge/shoulder or a designated stop area (pickup, emergency stop, parking approach).
	Turn (intersection/roundabout/U-turn)	Planned path onto a different road segment with a significant heading change; specify left/right.
	Lateral maneuver abort	Cancel an ongoing lateral maneuver (nudge, lane change, merge/split, pull-over) and re-center when safe.

Table 1: Closed-set driving decisions (longitudinal and lateral) used to anchor reasoning traces to explicit control intent. Annotators select at most one decision per channel (or *None*), ensuring decision-grounded supervision. Definitions emphasize operational intent and disambiguate visually or behaviorally similar maneuvers (e.g., *Lead obstacle following* vs. *Yield*, *Lane change* vs. *Merge / Split*). Each selected decision must be causally supported by evidence from the observed history window. LC denotes lane change.

clips per scenario to ensure dataset diversity. Detailed definitions of the scenarios are provided in Tab. 3.

Keyframe Labeling. Each raw clip contains 20 seconds of data and can generate multiple training samples, given the configuration of using a 2-second history to predict a 6-second future during both training and evaluation. Selecting keyframes for CoC annotation is therefore critical to maximizing the clarity of decision causality. For *reactive* scenarios, a keyframe is typically chosen by applying a short temporal buffer (approximately 0.5 seconds) before the ego vehicle initiates a behavior change corresponding to a driving decision. At this keyframe, the ego vehicle has accumulated sufficient observations within the preceding 2-second history to justify the forthcoming action, effectively avoiding casual confusion. Because the keyframe is positioned immediately prior to the decision-making moment, we ensure that a concrete driving decision is associated with the data sample, enabling the annotation of decision-grounded CoC traces. For *proactive* scenarios, we annotate a keyframe range: a time window during which the ego actively evaluates or prepares for a potential maneuver change. Detailed definitions of the keyframe or keyframe range for both reactive and proactive

Category	Example attributes to record (if decision-relevant)	Uncertainty
Critical objects	Type (veh./ped./cyclist/VRU), relative pose to ego (in-path, left/right, oncoming, crosswalk), motion (stopped, slowing, crossing, cut-in risk)	Low / High
Traffic lights	Current state (R/Y/G), arrow state, visibility/occlusion; presence of wait line	-
Yield/Stop control	Presence of signs, all-way vs two-way, stop/yield line location	-
Road events	Curvature/grade, speed bump, narrowing, roundabout, ramp/junction ahead	-
Lane / lanelines	Lane count, laneline type (dashed/solid), shoulder/bike lane, usable width	-
Routing intent	Target lane/turn (L/R/through), near-term split/merge, required lane for maneuver	-
ODD constraints	Weather/visibility, construction, emergency vehicles, school bus/rail rules	-

Table 2: Categories and example attributes of *critical components* that may serve as causal factors for driving decisions. Only those directly influencing the driving decision are labeled. Use a Low/High uncertainty tag when forecasting object behavior or when signals are partially occluded. The list is open-ended, allowing additional critical components to be added as needed.

scenarios are provided in Tab. 3. CoC reasoning traces are annotated only for samples corresponding to the keyframe timestamp or the keyframes sampled from the keyframe range.

4.3. Hybrid Labeling Procedure

To ensure both quality and scalability, we develop a hybrid labeling procedure that combines human labeling and auto-labeling. While auto-labels are sufficient for generating large-scale training data for reasoning VLA models, high-quality and human-verified data, on the order of $\sim 10\%$ of the total, is essential for further SFT, auto-label evaluation, and model evaluation. Our proposed hybrid labeling approach balances efficiency and accuracy, supporting both large-scale training and reliable model assessment.

4.3.1. Human Labeling

Two-Stage Labeling Procedure. Following the structured CoC described in Sec. 4.1, human annotators are required to complete a two-stage procedure designed to produce concise and causally grounded CoC write-ups.

1. **Stage I (0–2 s):** identify *critical components* from Tab. 2 within the observed history window (within 2s before the keyframe). This step helps prevent causal confusion by ensuring that only evidence available prior to the decision-making moment is considered. These critical components may influence the driving decision annotated in the next stage.
2. **Stage II (0–8 s):** (a) apply a safety exclusion filter to remove invalid data with illegal or unsafe driving behavior, (b) select the first post-keyframe driving decision for each channel (longitudinal and lateral; or *None*), (c) write a CoC reasoning trace that references only the causal factors identified in Stage I that lead to the driving decision, along with routing or regulatory signals when applicable.

To enforce a clear separation between Stage I and Stage II and minimize causal leakage, we designed a labeling tool that explicitly distinguishes historical video segments (0-2 s) from future segments (2-8 s). This tool also provides visual aids, including ego-dynamics plots (speed, acceleration, steering angle, and turn signals), BEV visualizations overlaid with lane topology, and obstacle bounding boxes in order to help annotators achieve a more accurate understanding of the driving scene.

Quality Assurance (QA). To maximize annotation quality and reduce potential bias, we implement a rigorous QA process. Each labeled instance first undergoes a quality check performed by a different annotator. Moreover, $10\% - 20\%$ of labeled instances are selected, based on the performance of the assigned annotators, for an additional auditing process conducted by a dedicated team of experienced auditors. Both the quality check and auditing process follow the same QA guidelines, with key rules summarized in Tab. 4. This QA process ensures that the desiderata of CoC are rigorously enforced while preserving flexibility for natural language expression.

Type	Scenario name	Keyframe Definition (Reactive) / Keyframe Range (Proactive)
Reactive	Slow for the lead vehicle	0.5 seconds before the ego decelerates behind a lead vehicle.
	Stop for the lead vehicle	Same as above.
	Stop for traffic light (TL) / traffic sign (TS)	Whichever occurs later: (1) 0.5 seconds before the ego begins to decelerate for a TL/TS; or (2) for a TL, the frame when it turns yellow/red.
	Resume at TL / TS	Whichever occurs later: (1) 0.5 seconds before the ego begins to accelerate from standstill due a TL/TS; or (2) for a TL, the frame when it turns green.
	Lane change (LC)	0.5 seconds before the ego starts to move off-center of its original lane.
	Yield to VRUs	0.5 seconds before the ego begins to decelerate or nudge for a VRU.
	Vehicle cut-in	Whichever occurs first: (1) when the contender signals a LC into ego's lane; or (2) when the contender starts to move off-center of its original lane for the LC if no blinker signal is given.
	Speed bump	0.5 seconds before the ego decelerates for the speed bump ahead.
	Nudge	0.5 seconds before the ego moves away from the lane center to avoid or give space to an obstacle.
	Bypass construction objects	0.5 seconds before the ego decelerates or nudges to construction objects or changes lane in response to construction objects modifying the lane.
	Risky driving	0.5 seconds before the ego decelerates, nudges or moves backward for a risky event or obstacle, e.g., lane-weaving leading vehicle, parked vehicle backing out, or oncoming vehicle crossing into ego's lane.
Proactive	Curvy road	Start: whichever occurs first - (1) 0.5 seconds before the ego begins to decelerate for the curve; or (2) when the ego enters the curve at current speed. End: when the ego exits the curve.
	Lane change (LC) preparation	Start: ego receives a reason to perform a LC (e.g., route or passing a slow lead) but cannot do it immediately due to a blocked target lane. End: Ego is ready to change lanes after gap searching or when traffic clears.
	Nudge preparation	Start: ego receives a reason to nudge for an obstacle, but cannot do it immediately due to traffic. End: Ego is ready to nudge once the traffic clears.
	Passing intersection	Start: ego enters the intersection when the front bumper crosses the stop line or crosswalk boundary. End: ego fully exits the intersection area.
	No yield to VRUs	Start: when VRUs appear with the intention to cross but are not yet crossing because (1) ego has the right of way, or (2) VRUs intentionally yield to ego. End: when the VRUs are no longer visible.

Table 3: Scenarios used in clip selection, along with keyframe and keyframe range definitions for CoC annotation. The goal is to identify critical reasoning moments within each selected clip, where a clear causal link can be established between observable factors and the ego vehicle's subsequent decision.

As a result, we generate high-quality CoC reasoning traces across diverse driving scenarios, with representative examples shown in Fig. 4.

4.3.2. Auto-Labeling

Keyframe Selection for Auto-Labeling. To efficiently scale up training data and enhance model generalization, we develop an auto-labeling pipeline for CoC annotation. To identify keyframes for auto-labeling, we first define a set of low-level meta actions and implement corresponding rule-based detectors to infer these meta actions at the frame level. Then, we treat the frame at which a meta action transition occurs as a decision-making moment, allowing us to determine the keyframe automatically and efficiently across large scale data.

Meta Actions. The complete list of these meta actions is provided in Tab. 5. These low-level meta actions are *atomic*, representing instantaneous kinematic changes in the ego vehicle's trajectory, and are therefore distinct from high-level driving decisions. A single high-level driving decision within a video segment typically consists of a sequence of such atomic meta actions across both longitudinal and lateral directions. For example, a left

Rule	Operational check
Causal coverage	Each selected decision references at least one Stage I component; otherwise mark *UNOBSERVED* with brief justification.
Causal correctness	Reasoning trace must logically explain the selected decision based on valid cause–and–effect relationships. Circular reasoning, misattributed causes, or missing necessary conditions are flagged for rework
Proximate cause	Prefer the immediate driver (e.g., stopped lead) over background conditions (e.g., red light when not first in queue).
Decision minimality	If no change in decision, label *None*.

Table 4: Quality assurance (QA) checklist for quality check and auditing process. Key rules tie closely to the desiderata of CoC: decision grounding, causal locality, and annotation economy.

Longitudinal		**Lateral**	
Gentle accelerate	Strong accelerate	Steer left	Steer right
Gentle decelerate	Strong decelerate	Sharp steer left	Sharp steer right
Maintain speed	Stop	Reverse left	Reverse right
Reverse	–	Go straight	–

Table 5: List of atomic meta actions defined for longitudinal and lateral directions. These meta actions represent instantaneous kinematic changes in low-level trajectories at the frame level, in contrast to high-level driving decisions that are composed of multiple atomic actions over a video segment.

lane-change decision may comprise a sequence of *steer left*, followed by a brief *steer right* to stabilize the vehicle heading, and then *go straight*, often accompanied by a *gentle accelerate* and *maintain speed*. For each 8-second data sample, we annotate at most one longitudinal and one lateral high-level driving decision, while atomic meta actions are automatically labeled at 10Hz.

Labeling Procedure. Next, we employ state-of-the-art VLMs such as GPT-5 (OpenAI, 2025) to perform offline auto-labeling through a multi-step reasoning process. This approach distills world knowledge from large models into structured CoC annotations, while balancing efficiency and cost. Similar to the human labeling pipeline, VLMs generate structured reasoning traces consisting of the identified driving decision, critical components, and a concise reasoning trace that links the driving decision to its causal factors. To support the reasoning process, the auto-labeling pipeline provides the model with both raw video and auxiliary signals, including the ego vehicle's trajectory, dynamic states, and meta actions. The video is sampled at 2 Hz to balance information density with the allowed input token budget within the auto-labeling model's context window.

To mitigate causal confusion, VLMs are prompted to use the 2-second historical video when identifying critical components. The subsequent 6-second future video, along with the ego's trajectories and meta actions, is then used to resolve multi-modality and determine the corresponding driving decision. During this process, the model ranks the importance of the identified causal factors and retains only those that directly influence the driving decision in the final reasoning trace.

4.3.3. Evaluation

Assessing open-ended text, especially reasoning traces, remains an open challenge in the AV research community, and evaluating causal-effect relationships in CoC introduces an additional layer of complexity. Prior datasets have typically relied on one of the following approaches:

(1) *Human evaluation* on a small subset of samples. While effective when labelers are properly guided, this approach is not scalable for large-scale evaluation or rapid iteration of labeling pipelines.
(2) *Heuristics-based metrics*, such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and CIDEr (Vedantam et al., 2015). These metrics focus on capturing only shallow text similarity and fail to reflect underlying causal reasoning, making them inadequate for evaluating our CoC dataset.

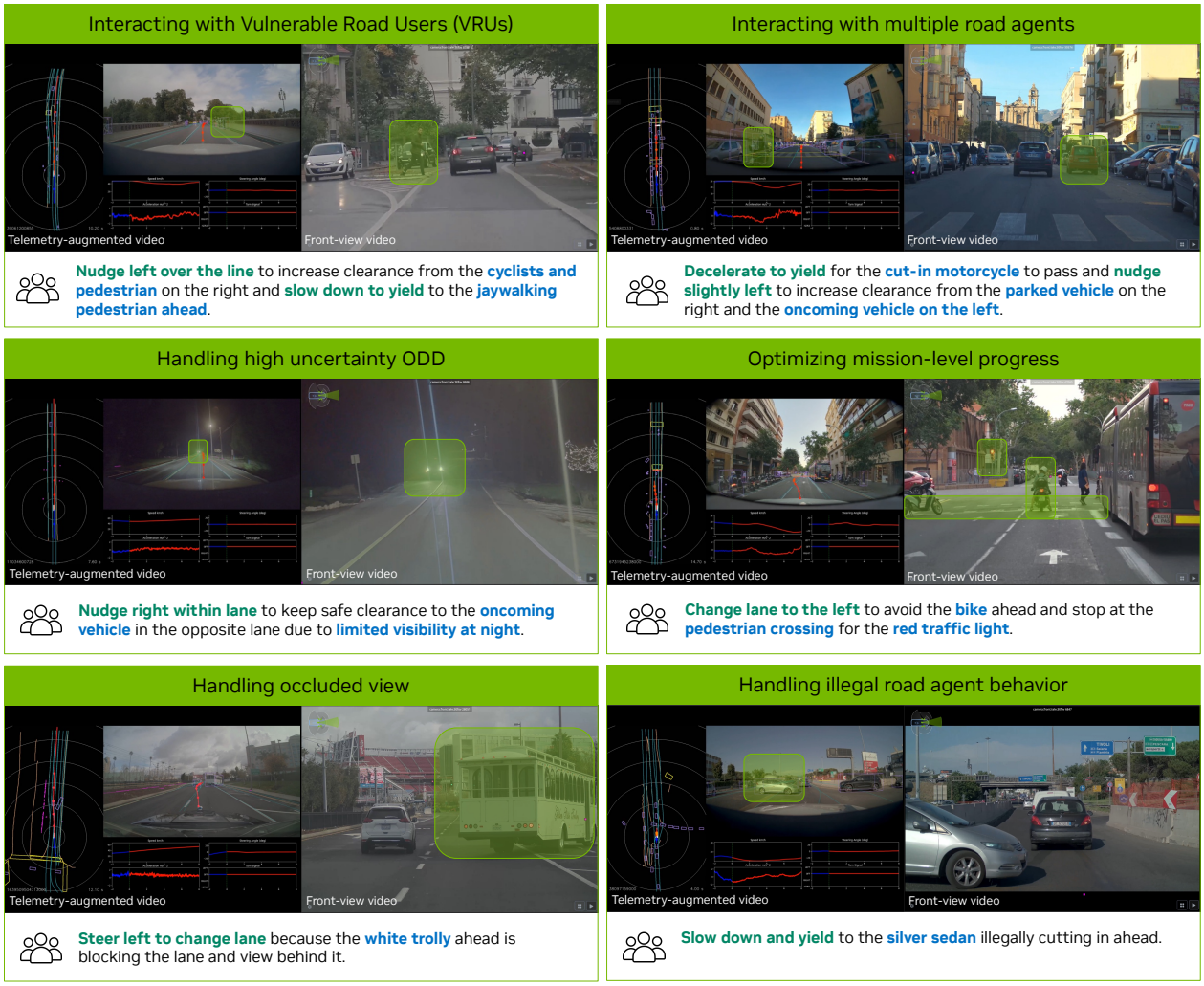

Figure 4: Examples of our labeled CoC reasoning traces, where **driving decisions** and **critical components** are organized into CoC and highlighted correspondingly.

(3) *LLM-based auto-evaluation*, which leverages LLMs' capacity to reason about causal relationships and scales effectively to large evaluation sets. However, LLMs are subject to hallucinations, particularly when assessing complex multi-step cause–and–effect chains.

Due to these challenges, prior works often lack a reliable method for reasoning dataset evaluation.

CoC Evaluation Procedure. To address these challenges, we adopt a hybrid evaluation strategy that combines human verification with LLM-based auto-evaluation. Specifically, we use GPT-5 (OpenAI, 2025) as an LLM evaluator and construct a curated evaluation set of 2K samples spanning representative scenarios listed in Tab. 3. To mitigate hallucination during LLM evaluation, we avoid using free-form text and grading results directly. Instead, we decompose the evaluation process into three structured subtasks covering driving decisions, presence of causal factors, and validity of the cause-and-effect relationship. By reformulating these aspects as a set of True/False questions, the evaluation process becomes more interpretable and better aligned with human judgment. To validate reliability, we compare LLM-based auto-evaluation against human evaluation on the same version of the auto-labeled dataset, and observe a 92% alignment rate, confirming the robustness of our LLM-based auto-evaluation. With this evaluation method, we find that the proposed structured CoC reasoning traces improve the causal relationship score by 132.8% relative to free-form reasoning traces, which do not enforce explicit driving decisions and critical components.

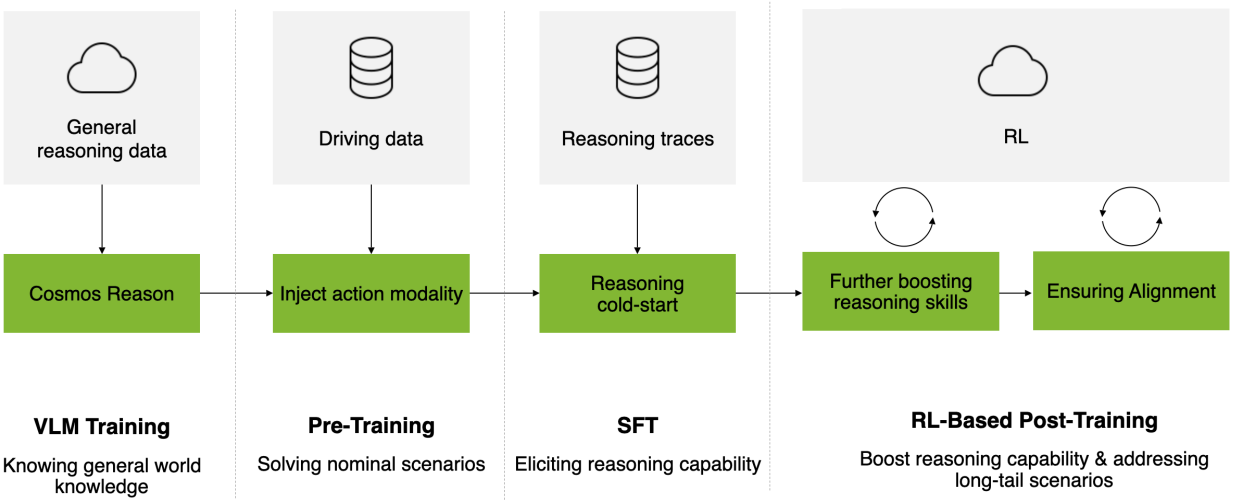

Figure 5: Overview of the Alpamayo-R1 model training pipeline, consisting of three key stages: (1) Action Modality Injection (Sec. 5.1), (2) Eliciting Reasoning (Sec. 5.2), and (3) RL-Based Post-Training (Sec. 5.3).

Effectiveness of Imperfect Auto-Labels. It is important to note that attaining a perfect (100%) score in causal-effect evaluation, were it even possible, is not a necessary condition for the usefulness of auto-labeled data. Given the inherent ambiguity of causal reasoning in complex driving scenarios, as well as noise in both human-labeled ground truth and evaluation metrics, it is unclear whether 100% agreement is a reasonable or well-defined target. Instead, the primary value of CoC's auto-labels lies in enabling large-scale SFT, which improves AR1's generalization across diverse driving scenarios. Empirically, as will be shown in Sec. 6, models trained on auto-labeled CoC traces already achieve significant improvements over baselines without reasoning supervision. Moreover, as will be described in Sec. 5, our training pipeline incorporates subsequent RL-based post-training steps which further strengthen reasoning capability and causal consistency. In parallel, as our human annotation effort scales, we plan to introduce additional rounds of SFT using human-labeled CoC reasoning traces, progressively improving causal grounding and interpretability.

5. Training Strategy

Building upon the Cosmos-Reason VLM backbone introduced in Sec. 3, which provides foundational physical reasoning capabilities through domain-specific SFT, we adopt a three-stage training strategy to transform the VLM into a reasoning-capable autonomous driving policy. As illustrated in Fig. 5, each stage progressively enhances distinct capabilities essential for robust and interpretable driving. In Sec. 5.1, we inject the action modality into the VLM by training with discrete trajectory tokens and adding an action-expert based on flow matching, enabling the model to predict vehicle control outputs. In Sec. 5.2, we improve the model's reasoning capability through SFT on the CoC dataset (Sec. 4), teaching the model to generate causally grounded explanations for better driving decisions. Finally, in Sec. 5.3, we employ RL with large reasoning model feedback to refine reasoning quality, align reasoning traces with executed actions, and optimize trajectory quality, producing interpretable and safe driving behavior.

5.1. Action Modality Injection

During training, we inject the action modality to the VLM through discrete tokens (Sec. 3.2.2) and train the VLM via cross-entropy loss over the training token sequence defined in Eq. (1). Following the control-based representation in Eq. (3), each trajectory consists of 64 waypoints with 2 quantized values per waypoint (acceleration a^i and curvature κ^i), resulting in 128 discrete tokens per trajectory. These are encoded with a set of special tokens dedicated to action representation. However, we do not use discrete trajectory tokens for inference, as detailed below.

Motivation for Dual Representation. The use of discrete tokens during training alongside a continuous flow-matching decoder at inference provides several key advantages. First, discrete tokenization enables *unified* autoregressive training in which reasoning and trajectories share a common token space, allowing the VLM to tightly couple causal explanations with vehicle behaviors through standard next-token prediction. Second, discrete representations facilitate RL optimization by allowing direct gradient flow during post-training (Sec. 5.3), allowing policy gradient methods such as GRPO (Shao et al., 2024) to jointly refine reasoning quality and reasoning-action consistency. Third, the discrete representation provides strong supervision for learning vehicle dynamics, while the flow-matching expert ensures physically feasible and multi-modal outputs. Finally, flow-matching decoding offers computational efficiency, generating continuous trajectories substantially faster than autoregressively sampling 128 discrete tokens, enabling real-time inference.

Similar to $\pi_{0.5}$-KI (Driess et al., 2025), we adopt a separate action-expert to decode actions via flow matching (Janner et al., 2022; Lipman et al., 2023; Zhong et al., 2023; Jiang et al., 2023). The action-expert follows the same Transformer architecture as the VLM, using the same number of attention heads and attention dimensions, but with a smaller hidden embedding size and MLP dimension for efficiency. At each diffusion timestep t in the diffusion schedule, the action-expert takes as input both the KV-cache from the sequence $[\boldsymbol{o}_{\text{image}}, \boldsymbol{o}_{\text{egomotion}}, \text{REASON}]$ in the VLM and the embedded representation of the noisy control $\boldsymbol{a}_t$ (with the diffusion time t also embedded and added to the feature). The expert then predicts the vector field $\mathbf{v}_\Theta(\boldsymbol{a}_t, \boldsymbol{o}, \text{REASON})$ by projecting the final layer feature through an MLP head, where Θ denotes the learnable parameters. We train the action-expert using a vanilla conditional flow matching loss (Lipman et al., 2023),

$$L_{\text{cfm}}(\Theta) = \mathbb{E}_{t \in p_{\text{schedule}}, (\boldsymbol{o}, \text{REASON}) \in \mathcal{D}_{\text{data}}} \|\mathbf{v}_\Theta(\boldsymbol{a}_t, \boldsymbol{o}, \text{REASON}) - \mathbf{u}(\boldsymbol{a}_t | \boldsymbol{a})\|. \tag{6}$$

In practice, we adopt the Gaussian conditional optimal transport (OT) path and sample $\boldsymbol{a}_t = t\boldsymbol{a} + (1-t)\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where the target vector field admits a closed-form expression:

$$\mathbf{u}(\boldsymbol{a}_t | \boldsymbol{a}) = \boldsymbol{a} - \boldsymbol{\epsilon}. \tag{7}$$

During inference, starting with $\boldsymbol{a}_0 \in \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we perform denoising through Euler integration:

$$\boldsymbol{a}_{t+\delta_t} = \boldsymbol{a}_t + \delta_t \, \mathbf{v}_\Theta(\boldsymbol{a}_t, \boldsymbol{o}, \text{REASON}). \tag{8}$$

By default, we use $\delta_t = 0.1$ during inference and set p_{schedule} to a shifted beta distribution during training, as suggested by Physical Intelligence et al. (2025). During training, we apply a stop-gradient to the KV-cache produced by the VLM to prevent gradients from the expert back-propagating into the VLM weights.

5.2. Eliciting Reasoning

Having established a VLA with action generation capabilities in Sec. 5.1, the next challenge is to enable the model to perform structured and causally grounded reasoning that explains *why* specific driving decisions are made. This capability is critical for handling complex, safety-critical scenarios where pure pattern matching from imitation learning may fail (Wei et al., 2022). To achieve this, we leverage the structured CoC dataset introduced in Sec. 4, which provides decision-grounded and causally linked reasoning traces paired with expert trajectories. We perform SFT on the CoC dataset to teach the model to generate reasoning traces through imitation, where each reasoning trace is anchored to explicit driving decisions (Tab. 1) and grounded in critical scene components (Tab. 2). While SFT enables the model to scaffold basic reasoning capabilities, we further refine reasoning quality and enforce reasoning-action consistency through RL in Sec. 5.3. Formally, each training sample consists of a multi-camera driving scene observation $\boldsymbol{o} = [\boldsymbol{o}_{\text{image}}, \boldsymbol{o}_{\text{egomotion}}]$, a structured CoC reasoning trace REASON that explains the causal factors behind the ego vehicle's decision, and the corresponding ground-truth control-based trajectory representation $\boldsymbol{a}$ defined in Eq. (3). Following the sequence formulation in

Eq. (1), the SFT objective maximizes the conditional log-likelihood of the reasoning–action sequence:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(\boldsymbol{o},\text{REASON},\boldsymbol{a})\sim\mathcal{D}_{\text{CoC}}}\left[\log \pi_\theta(\text{REASON}, \boldsymbol{a} \mid \boldsymbol{o})\right], \tag{9}$$

where π_θ denotes the VLA policy parameterized by θ, encompassing the vision encoder, language backbone, and corresponding embedding adapters. In practice, we apply the cross-entropy loss over both the reasoning tokens and the discrete trajectory tokens (128 tokens per trajectory as described in Sec. 5.1), enabling the model to learn the joint distribution of language-based reasoning and action prediction in a unified autoregressive framework.

Why SFT Alone is Insufficient. This imitation learning stage allows the model to internalize human-like reasoning patterns: learning not only *what* action to take, but also *why* such actions are appropriate given specific visual and contextual cues. As shown in Fig. 8, SFT on CoC data already yields measurable improvements in trajectory prediction accuracy compared to models trained without explicit reasoning supervision. However, while SFT enables the VLA model to scaffold reasoning traces, it remains inherently limited by several factors:

(1) *Data bias and annotation noise*: Auto-labeled data may contain imperfect causal relationships (Sec. 4.3.1), causing the model to overfit to annotation artifacts rather than learning robust causal reasoning.

(2) *Limited generalization*: The model may memorize common reasoning patterns without developing deeper causal understanding, failing to generalize to novel scenarios.

(3) *Weak visual grounding*: Next-token prediction does not enforce visual consistency; the model may hallucinate causal factors not present in the scene (Fig. 10).

(4) *Reasoning–action inconsistency*: Joint optimization does not explicitly enforce alignment between stated reasoning and predicted trajectories, potentially leading to contradictory explanations (Fig. 11).

To address these limitations, we leverage RL with large reasoning model feedback and explicit reasoning-action consistency rewards, as described in Sec. 5.3.

5.3. RL-based Post-Training

As discussed in Sec. 5.2, while supervised fine-tuning on CoC data enables the model to generate structured reasoning traces, it remains inherently limited by data bias, generalization challenges, weak visual grounding, and reasoning-action inconsistency. To address these limitations, we introduce an RL-based post-training framework shown in Fig. 6 that optimizes three complementary reward signals: reasoning quality (via large reasoning model feedback), reasoning-action consistency, and trajectory quality. This approach directly tackles the shortcomings of SFT by providing targeted feedback that evaluates both the causal correctness of reasoning and its alignment with executed actions.

5.3.1. Post-Training Algorithm

Large-scale foundation model post-training has emerged as a central strategy to enhance the reasoning capabilities and generation quality of large-scale foundation models (Christiano et al., 2017; DeepSeek-AI, 2025). Recently, these techniques have been extended to the embodied AI domain, encouraging VLA models to generate actions that better reflect human intent across diverse embodiments, including autonomous driving (Tian and Goel, 2025) and generalist robotic agents (Tian et al., 2024; Zhang et al., 2025). In our *reasoning* VLA context, the alignment stage extends beyond improving motion generation; it explicitly enhances reasoning quality grounded in embodied settings and enforces reasoning–action consistency, both of which are key properties for achieving interpretable and trustworthy autonomy.

We adopt GRPO (Shao et al., 2024) as our alignment algorithm. GRPO extends standard policy gradient methods by optimizing relative advantages within a group of sampled model rollouts rather than relying on absolute reward signals. Specifically, given a group of model rollouts $\{\tau_i\}_{i=1}^{K}$ sampled from the current model

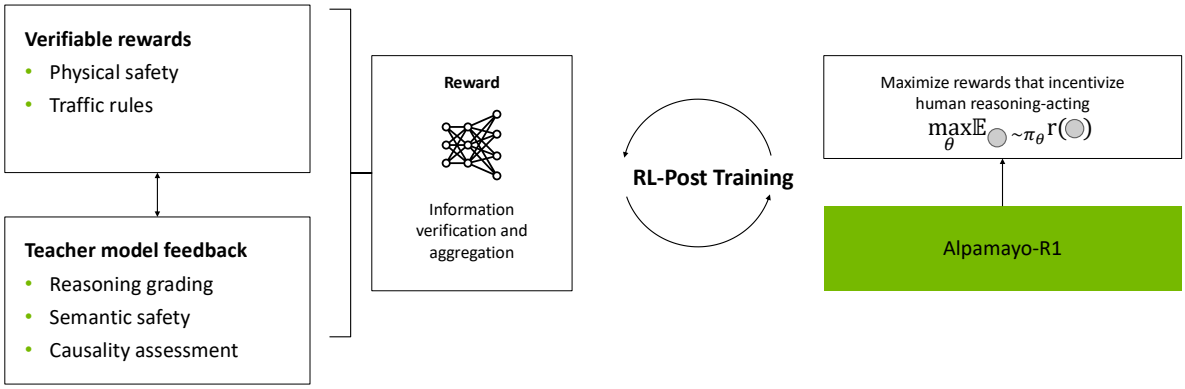

Figure 6: Overview of our RL-based post-training framework. We optimize three reward components: reasoning quality (via large reasoning model feedback), reasoning-action consistency, and trajectory quality, to align the model's generated reasoning traces with its predicted actions.

π_θ, each with an associated scalar reward r_i, the objective of GRPO is defined as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}\tau_i \sim \pi_\theta \left[\frac{\exp(\beta A_i)}{\sum_j \exp(\beta A_j)} \left(\log \pi_\theta(\tau_i) - \lambda_{\text{KL}} \text{KL}[\pi_\theta(\tau_i) \| \pi_{\text{ref}}(\tau_i)] \right) \right], \quad A_i = r_i - \bar{r}. \tag{10}$$

Here, A_i denotes the relative advantage of each trajectory within the group, $\bar{r}$ is the group-average reward, and β controls the sharpness of the weighting distribution. The KL regularization term with coefficient λ_{KL} penalizes deviations from the reference policy π_{ref} (typically the SFT model), preventing over-optimization on noisy or biased reward signals and preserving linguistic and behavioral priors learned during pre-training.

5.3.2. Reward Model

Our reward model integrates three complementary signals that together evaluate both what the model reasons and how it acts. Specifically, the total reward r for each rollout is composed of three components: reasoning quality reward, reasoning-action consistency, and low-level trajectory quality.

Grading Reasoning with Large Reasoning Models. To mitigate the issue where reasoning traces can exhibit hallucinations that produce plausible yet unsafe or causally inconsistent plans, we employ large reasoning models (LRMs) as automatic evaluators to provide scalable, high-quality feedback on reasoning quality. Inspired by recent advances in LLM alignment, where expert models serve as judges to provide scalable feedback (Bai et al., 2022; Lee et al., 2023), we leverage state-of-the-art LRMs (e.g., DeepSeek-R1 (DeepSeek-AI, 2025), Cosmos-Reason (NVIDIA et al., 2025)) as *reasoning critics* to evaluate the quality of reasoning traces generated by the VLA. We choose an LRM as the critic because, although such models may struggle to generate driving-specific reasoning due to limited embodiment priors, they exhibit strong verification and evaluation capabilities. In other words, even when generation in this domain is imperfect, their ability to assess logical soundness, causal alignment, and contextual consistency remains highly reliable (also known as the generation–verification gap (Song et al., 2024)). The resulting reward signal provides a continuous measure of reasoning quality, enabling RL to iteratively refine the model's ability to generate grounded and logically consistent reasoning.

Reasoning Critic Design. For each training sample, the LRM critic takes as input the multi-camera visual observation $\boldsymbol{o}_{\text{image}}$ at the last frame of the 2-second history window, the ground-truth CoC reasoning trace $\text{REASON}_{\text{GT}}$ from the dataset, and the model-generated reasoning trace $\text{REASON}_{\text{pred}}$ produced by the current policy π_θ. The critic evaluates how well $\text{REASON}_{\text{pred}}$ aligns with $\text{REASON}_{\text{GT}}$ along two dimensions: *behavior consistency*, whether the predicted reasoning describes a driving decision consistent with ground truth; and *causal reasoning quality*, whether it correctly identifies causal factors observable in the scene's history according

to CoC principles (Sec. 4.1). The critic grades the predicted reasoning according to a structured rubric focused on behavior consistency and causal reasoning consistency:

Prompt : LLM Reasoning Grading Rubric

You are an expert evaluator for autonomous driving reasoning traces. The reasoning trace describes what the ego vehicle should be doing and the reasons and factors that lead to the behavior. Your task is to score how well a predicted reasoning trace (`PRED`) aligns with the ground truth (`GT`) in terms of behavior consistency and causal reasoning.
Scoring rubric (0–5):

- 5 Behavior & causal reasoning fully consistent.
- 4 Behavior correct; causal reasoning mostly consistent.
- 3 Behavior roughly correct, but incomplete or slightly incorrect reasoning.
- 2 Behavior partially incorrect or reasoning largely inconsistent.
- 1 Behavior is wrong or contradicts GT.
- 0 Completely unrelated or opposite.

The resulting scalar score r_{reason} is used as the reasoning reward. This signal encourages the model to generate reasoning traces that not only describe correct driving behaviors but also maintain causal fidelity, accurately explaining why an action is taken based on visual context and traffic cues.

CoC-Action Consistency. To ensure that the model's action generation faithfully follows its reasoning, we introduce a CoC–action consistency reward that measures behavioral alignment between the generated reasoning trace and the corresponding predicted ego trajectory. Specifically, for each reasoning–action rollout, we convert the predicted motion trajectory into a sequence of meta-actions (interpretable motion primitives) described in Sec. 4.3.1. These meta-actions encode the ego vehicle's control behavior along both the longitudinal (acceleration/braking) and lateral (steering) directions. We then parse the generated reasoning trace to infer the ego's intended behavior and compare it against the meta-actions derived from the predicted trajectory using rule-based matching. If the described behavior in the reasoning trace and the meta-action are consistent across both axes, we assign $r_{\text{consistency}} = 1$; otherwise, $r_{\text{consistency}} = 0$. In cases where the reasoning cannot be parsed into a valid driving decision (i.e., the intent is not recognized within the closed decision set used for auto-labeling), we conservatively assign $r_{\text{consistency}} = 0$. Although based on simple rule-based logic, this binary reward plays a crucial role in improving the trustworthiness of the model's reasoning–action coupling. By explicitly penalizing inconsistencies and rewarding only correct matches, it encourages the model to generate reasoning that not only sounds plausible but also translates into coherent, physically consistent behavior.

Low-Level Trajectory Quality. To ensure that the generated motion trajectories remain physically feasible, comfortable, and safe to execute, we include a low-level trajectory quality reward that evaluates the model's motion outputs in continuous space. This component complements the above reasoning- and consistency-level rewards by directly regularizing the trajectory's physical properties. The reward combines three terms:

$$r_{\text{traj}} = \lambda_{\text{L2}} \|x_{\text{pred}} - x_{\text{expert}}\|_2^2 + \lambda_{\text{coll}} \mathbb{I}[\text{collision}(x_{\text{pred}})] + \lambda_{\text{jerk}} J(x_{\text{pred}}), \tag{11}$$

where x_{pred} and x_{expert} denote the predicted and expert trajectories, respectively; $\mathbb{I}[\text{collision}(x_{\text{pred}})]$ is a binary indicator that denotes whether the predicted motion leads to a collision with surrounding obstacles; and $J(x_{\text{pred}})$ measures the magnitude of the jerk to penalize abrupt or uncomfortable motion. The L2 imitation term encourages proximity to expert demonstrations, promoting stable learning and smooth driving profiles. The collision penalty ensures safety, while the jerk regularization improves comfort and control smoothness. Together, these terms anchor the learning of the model to human-like, safe, and comfortable motion, reinforcing the physical plausibility of the trajectories generated during the alignment process.

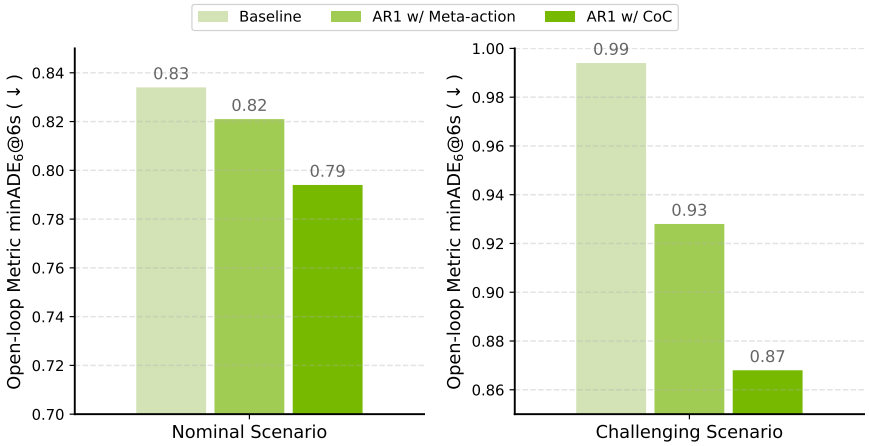

Figure 7: Compared to models that only output trajectories or only output meta-actions and trajectories, Alpamayo-R1 achieves improvements in both nominal and challenging scenarios.

5.3.3. Post-Training Data Curation for Cost-Effective Training

RL–based post-training is computationally expensive due to its iterative nature: each policy update requires multiple model rollouts, reward evaluations, and gradient steps across large batches of reasoning and trajectory samples. Moreover, unlike the SFT stage where the loss is directly computed from labeled data, our post-training procedure involves on-policy sampling and LRM-based reward function calls, which amplify both compute and data costs. Consequently, scaling RL to the full pre-training data would be prohibitive in both training time and compute resources. To address this, we curate a high-information-gain dataset for RL post-training. The key idea is to prioritize samples where the model's implicit reward signal (encoded in its logits) disagrees with the explicit reward model.

Specifically, for each sample, we compute the model's predicted probability distribution derived from its logits, and the corresponding probability distribution implied by the rewards, which we obtain by transforming the reward into a Boltzmann distribution $p_{\text{reward}}(\tau_i) = \frac{\exp(\beta\, r_i)}{\sum_j \exp(\beta\, r_j)}$. A large divergence between these two distributions indicates that the model's internal preference (its implicit reward) conflicts with the externally defined reward signal. Such disagreement reveals samples where the model's learned reward is inaccurate, making them particularly valuable for alignment. We therefore prioritize these high-disagreement samples to construct a focused post-training dataset, while mixing in a similar proportion of randomly sampled data to preserve distributional diversity and stabilize training. By focusing RL updates on this hybrid set, we achieve both high alignment efficiency and robust learning dynamics compared to uniformly sampled data.

5.3.4. Post-Training Infrastructure

To conduct our RL experiments, we develop a customized version of the Cosmos-RL framework (NVIDIA, 2025) that is specifically designed for AV reasoning tasks. This system provides a scalable, modular infrastructure for large-scale multimodal RL and fits directly with other parts of the Alpamayo-R1 system. It supports distributed data loading, mixed-parallelism training, vLLM-based rollout generation (Kwon et al., 2023), and reward computation across multiple GPU nodes, enabling efficient, high-throughput policy optimization.

6. Experiments

We conduct comprehensive evaluations of Alpamayo-R1 across multiple dimensions to assess its reasoning capabilities, trajectory prediction accuracy, and closed-loop driving performance. We first highlight in Fig. 7 that the proposed Alpamayo-R1 significantly outperforms the trajectory-only baseline, particularly in challenging scenarios that intuitively require complex reasoning to make better driving decisions.

Table 6: Open-loop evaluation of models on the CoC dataset. The base model is pre-trained with $\mathcal{D}_{\text{overall}}$ and all other models are finetuned on the CoC dataset, then evaluated on held-out CoC test data. Numbers with green background are the best under each setting.

ID	Model Name	Route	Parameters	minADE_6@3s↓	minADE_6@6s↓
1	Base model (action modality)	×	0.5B	0.284	0.996
2	+ Ft. w/ Traj.	×	0.5B	0.282	0.971
3	+ Ft. w/ Meta-action & Traj.	×	0.5B	0.291	0.988
4	+ Ft. w/ CoC & Traj. (AR1)	×	0.5B	0.279	0.955
5	Base model (action modality)	×	3B	0.291	0.977
6	+ Ft. w/ Traj.	×	3B	0.293	0.976
7	+ Ft. w/ Meta-action & Traj.	×	3B	0.280	0.927
8	+ Ft. w/ CoC & Traj. (AR1)	×	3B	0.275	0.908
9	Base model (action modality)	✓	0.5B	0.264	0.848
10	+ Ft. w/ Traj.	✓	0.5B	0.262	0.834
11	+ Ft. w/ Meta-action & Traj.	✓	0.5B	0.264	0.821
12	+ Ft. w/ CoC & Traj. (AR1)	✓	0.5B	0.254	0.794

Table 7: Open-loop evaluation of models on the challenging dataset. All models are finetuned on the CoC dataset and evaluated on the challenging dataset.

ID	Model Name	Route	Parameters	minADE_6@3s↓	minADE_6@6s↓
1	Ft. w/ Traj.	✓	0.5B	0.315	0.994
2	Ft. w/ Meta-action & Traj.	✓	0.5B	0.301	0.928
3	Ft. w/ CoC & Traj. (AR1)	✓	0.5B	0.290	0.868

In the following sections, we first present the evaluation protocol in Sec. 6.1. Next, we illustrate how our reasoning-capable model contributes to an improved driving policy in Sec. 6.2. In Sec. 6.3 we further demonstrate the improvements in behavioral alignment achieved through RL. From Sec. 6.4 to Sec. 6.6 we conduct a comprehensive ablation study on the backbone model, the trajectory expert model, and the vision encoder to gain deeper insight into the effectiveness of our proposed methodology. Finally, we present an on-vehicle demonstration showcasing the real-world performance of our model.

6.1. Evaluation Protocol

Our evaluation strategy consists of four complementary components:

(1) *open-loop trajectory prediction* on both nominal and long-tail driving scenarios to measure planning accuracy;
(2) *closed-loop simulation* using AlpaSim (NVIDIA et al., 2025) to assess safety and robustness when the model controls the vehicle in realistic scenarios;
(3) *ablation studies* examining the impact of key architectural choices, including vision-language model scaling, vision encoding strategies, reasoning integration, and action decoding strategies;
(4) *on-vehicle road tests* to validate real-world deployment of the model in autonomous driving scenarios.

Dataset. We train and evaluate models on internal driving data collected across diverse geographic regions in the US and EU, with all evaluation data strictly geo-fenced and held out from training regions to prevent information leakage. Our evaluation encompasses both nominal driving scenarios in dataset $\mathcal{D}_{\text{overall}}$ and challenging long-tail cases in $\mathcal{D}_{\text{hard}}$ to thoroughly test the model's ability to handle rare, safety-critical events. In detail, the full training and evaluation dataset comprises 80,000 hours of driving data collected from multiple ego-vehicles operating in more than 1,700 cities in 25 countries. It encompasses diverse driving scenarios,

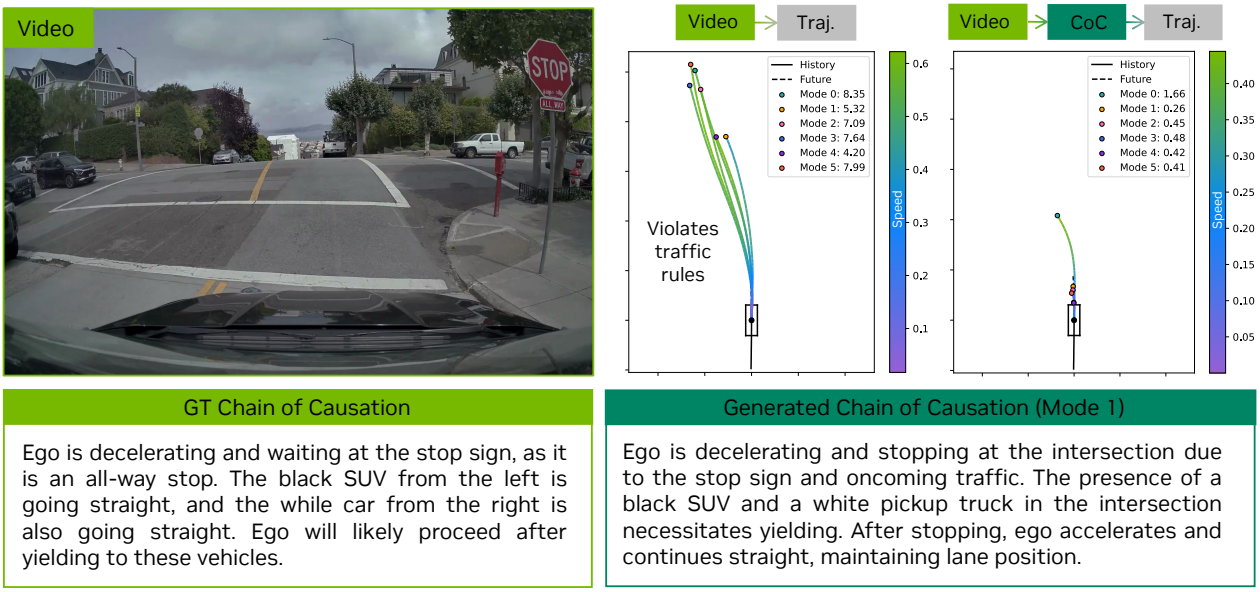

Figure 8: Policy improvements via eliciting reasoning: Alpamayo-R1 generates a correct reasoning trace at an all-way stop sign intersection and yields to other vehicles that enter the intersection earlier than ego.

including highway and urban environments, under various weather conditions, times of day, and traffic densities. The raw sensory inputs consist of video recordings from a surround-view seven-camera setup, accompanied by precise camera calibration parameters and ego-motion data. For model training and evaluation, we use two front-facing cameras as input: a front wide-angle camera with 120° field of view and a front telephoto camera with 30° field of view, providing complementary perspectives for both near-field and far-field scene understanding.

In addition to the general driving dataset $\mathcal{D}_{\text{overall}}$, we construct the CoC dataset (Sec. 4) consisting of 700K video segments with structured CoC. This dataset is used for fine-tuning models to elicit reasoning capabilities (Sec. 6.2) and for RL-based post-training alignment (Sec. 6.3).

Open-Loop Evaluation. For open-loop trajectory prediction, we evaluate models over a prediction horizon of 6 seconds, corresponding to the ego-vehicle's planned waypoints. We use minADE and ADE as the evaluation metric. minADE is computed over 6 samples (minADE_6) and is defined as the minimum distance between the ground-truth future trajectory and the best-matching trajectory among 6 predictions generated by the model. ADE (Average Displacement Error) is the average distance between the predicted trajectory and the ground-truth trajectory across all future timesteps.

Closed-Loop Evaluation. It is well established that strong open-loop results do not necessarily translate into reliable closed-loop driving performance (Dauner et al., 2023). To address this gap, we further evaluate our models within *AlpaSim* (NVIDIA et al., 2025), an open-source closed-loop end-to-end simulator based on state-of-the-art neural reconstruction technology (Wu et al., 2025). AlpaSim leverages a temporal 3D Gaussian Splatting representation from recorded real-world driving logs and, during closed-loop evaluation, uses it to synthesize novel viewpoints when the ego vehicle deviates from the recorded trajectory. During evaluation, predicted trajectories are tracked by a model predictive controller (MPC), and vehicle dynamics follow a dynamically extended bicycle model. Traffic agents, including vehicles and pedestrians, follow their recorded trajectories.

We evaluate models in 75 challenging 20-second scenarios, selected for their dense ego–agent and agent–agent interactions. While this may appear as a limited set, these scenarios are specifically curated to represent the most demanding safety-critical situations requiring complex reasoning and interactive decision-making. We

report the following AlpaSim metrics:

(1) *offroad rate*: percentage of scenarios where the ego vehicle drives outside of the drivable area;
(2) *close encounter rate*: percentage of scenarios where the ego vehicle experiences a close encounter with another traffic agent;
(3) *AlpaSim score*: average distance driven in km between events, where events correspond to offroad or close encounter occurrences;
(4) *AlpaSim score at fault*: same as AlpaSim score but considering only close encounters where the ego vehicle is deemed responsible, i.e., excluding rear-end close encounters.

The simulation ends after the first close encounter or off-road event. To mitigate rendering artifacts, events in which the ego deviates more than 4 m from the original recorded trajectory are excluded from all metric computations.

6.2. Policy Improvements from Reasoning

One of the key contributions of this work is the use of the proposed CoC data to improve driving policies. To evaluate the impact of reasoning on driving performance, we start with a base model pre-trained on $\mathcal{D}_{\text{overall}}$ with action modality injection (Sec. 5.1), then fine-tune it on the CoC dataset with different reasoning modalities: meta-action descriptions and full chain-of-causation reasoning traces. During inference, models trained with CoC reasoning generate explicit reasoning outputs alongside trajectory predictions, enabling them to better handle challenging scenarios that require multi-step decision making. We compare three fine-tuning strategies: (1) trajectory prediction only, (2) meta-action and trajectory prediction, and (3) chain-of-causation reasoning and trajectory prediction (Alpamayo-R1). All models are evaluated on held-out CoC test data in two settings: with and without route information provided to the model.

Open-Loop Improvements. As shown in Tab. 6 (nominal scenarios) and Tab. 7 (challenging scenarios), incorporating CoC reasoning yields substantial improvements in open-loop trajectory prediction in both settings. Without route information, AR1 achieves a minADE_6 of 0.955m at 6s, a 4.1% improvement over the base model and outperforming both trajectory-only (0.971m) and meta-action (0.988m) baselines. With route information, the gains are more pronounced: AR1 achieves 0.794m, representing 4.8% improvement over the trajectory-only baseline (0.834m). Scaling to 3B parameters further improves performance, with AR1-3B achieving 0.908m (without route), demonstrating the benefits of increased model capacity for complex reasoning tasks. In challenging scenarios, the improvements are even larger, with AR1 achieving 0.868m, a *12% improvement* over the trajectory-only baseline (0.994m), demonstrating the model's enhanced capability to handle safety-critical long-tail events through structured reasoning.

These results demonstrate that explicit reasoning capabilities enable the model to more effectively leverage contextual information such as route guidance and handle complex driving scenarios that require anticipating future interactions. Fig. 8 illustrates qualitative examples where the CoC-enabled model successfully generates correct reasoning traces and yields to vehicles in challenging scenarios, while baseline models fail to anticipate these interactions.

Closed-Loop Improvements. As shown in Tab. 8, AR1 achieves a 35% reduction in off-road rate (11% vs 17%) and 25% reduction in close encounter rate (3% vs 4%) compared to the trajectory-only baseline. The overall AlpaSim score improves from 0.38 to 0.50, demonstrating that reasoning-based decision making improves safety in dynamic closed-loop scenarios. Fig. 9 presents two qualitative examples that demonstrate that our model can successfully perform closed-loop driving in challenging scenarios within AlpaSim.

6.3. RL Post-Training for Improving Reasoning, Consistency, and Safety

While SFT on CoC data enables the model to jointly generate reasoning traces and actions, it does not guarantee that these traces are causally grounded or that the resulting actions faithfully reflect the reasoning or align

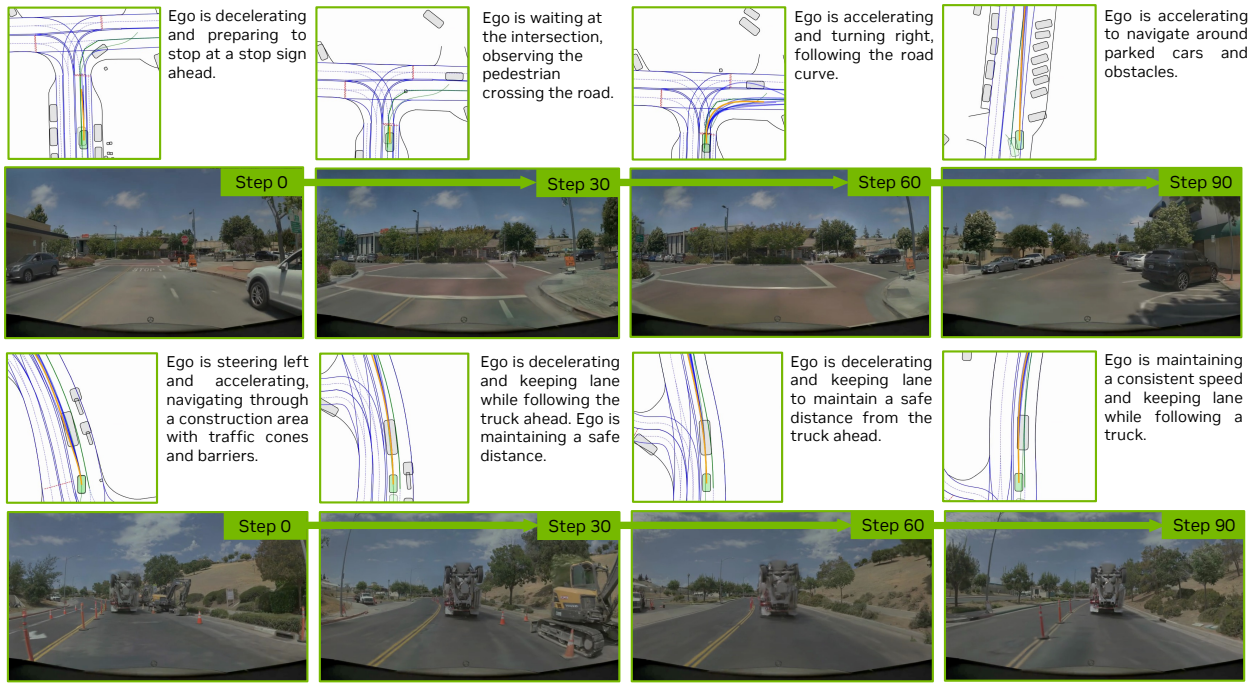

Figure 9: Examples of closed-loop evaluation in AlpaSim (NVIDIA et al., 2025). The top row presents an intersection scenario, whereas the bottom row illustrates a construction scenario.

Table 8: Closed-loop evaluation results in AlpaSim (NVIDIA et al., 2025). All models are evaluated without route information across 75 challenging scenarios. Baseline refers to the trajectory-only model fine-tuned on CoC training data without reasoning.

Model	Off-Road Rate ↓ (%)	Close Encounter Rate ↓ (%)	AlpaSim Score ↑	AlpaSim Score (at fault) ↑
Baseline	17.0±3.0	4.0±3.0	0.38±0.04	0.86±0.11
Alpamayo-R1	11.0±2.0	3.0±2.0	0.50±0.08	0.87±0.18

with human driving norms. To address this gap, we apply RL-based post-training to simultaneously improve reasoning quality, reasoning-action consistency, and trajectory quality (see Sec. 5.3 for methodology details). In this section, we post-train a 0.5B AR1 model fine-tuned on CoC data, and demonstrate the impact of different reward components on model behavior.

The Value of Learning from LRM Feedback. To ensure that the model's reasoning traces are not only fluent but also causally grounded and contextually accurate, we introduce a reasoning reward derived from LRM feedback (more details are in Sec. 5.3). This reward provides a continuous evaluation signal that measures the logical consistency and causal correctness of each generated reasoning trace with respect to the driving scene. Specifically, the average reasoning score of the most-likely rollout among six generations improves by approximately 45% (3.1→4.5) when the reasoning reward is applied. In Fig. 10, we illustrate two qualitative examples showcasing the model's behavioral differences before and after post-training. In the left scenario, the ego vehicle approaches a construction site. The most-likely mode generated by the SFT-pretrained model overlooks the construction barriers and describes the scene as a normal driving situation, failing to recognize the need for evasive behavior. After post-training, however, the model's reasoning correctly attends to the construction area and explains that the ego vehicle should nudge right to avoid obstacles. Similarly, in the right scenario, two pedestrians are about to clear the path. The most-likely mode generated by the SFT-pretrained model overlooks this contextual cue and fails to anticipate that the ego vehicle should prepare to accelerate. After post-training, the model correctly recognizes that the pedestrians are exiting the drivable area and reasons

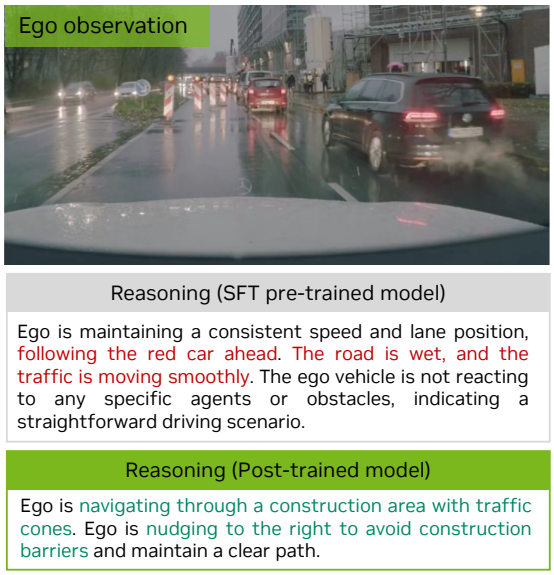

Figure 10: Post-training with the reasoning reward improves causal understanding and contextual reasoning in driving scenarios. **Left**: The base model overlooks construction barriers and fails to initiate evasive action, while the post-trained model correctly reasons that the ego should nudge right to avoid obstacles. **Right**: The base model misses that pedestrians are clearing the path, whereas the post-trained model correctly reasons that it is safe for the ego vehicle to accelerate.

Table 9: Improvements from RL-based post-training. We evaluate the impact of RL-based post-training on the model's reasoning, consistency, and motion quality. Metrics are computed from the most-likely rollout among six generated rollouts to assess how RL alignment influences the model's generation distribution. We measure ADE, reasoning quality graded by the large reasoning critic (Sec. 5.3.2), reasoning–action consistency, and close encounter rate. Evaluations are conducted on the full CoC dataset introduced in Sec. 6.2. We compare four configurations: the SFT-pretrained base model and three RL post-training variants incorporating different combinations of reasoning, consistency, and safety rewards.

Training strategy	ADE ↓	Reasoning Grading ↑	Reasoning–Action Consistency Score ↑	Close Encounter Rate (%) ↓
SFT	2.12m	3.1	0.62	6.9
SFT + RL (r_{reason})	2.19m	4.5	0.53	5.8
SFT + RL ($r_{\text{reason}} + r_{\text{consistency}}$)	1.92m	4.5	0.85	6.2
SFT + RL ($r_{\text{reason}} + r_{\text{consistency}} + r_{\text{safety}}$)	1.94m	4.4	0.83	3.7

that it is safe for the ego vehicle to resume motion.

The Value of Enforcing Reasoning-Action Consistency. Interestingly, when the post-training stage optimizes solely for the reasoning reward, the reasoning score indeed improves; however, both the ADE metric and reasoning–action consistency degrade compared to the base model. This indicates that optimizing for reasoning quality alone can lead to ungrounded or overconfident reasoning, where the model produces fluent but causally disconnected explanations that fail to translate into coherent actions. The consistency reward is therefore crucial for anchoring reasoning to physically realizable behaviors, ensuring that improvements in interpretability do not come at the expense of control fidelity. Specifically, when jointly optimizing both the reasoning and consistency rewards, the post-trained model achieves a 9.4% reduction in most-likely mode ADE (2.12m→1.92m), a 45% improvement in the reasoning score (3.1→4.5), and a 37% increase in reasoning–action consistency (0.62→0.85). These results demonstrate that the two reward components are complementary: the reasoning reward enhances interpretability and causal grounding, while the consistency reward ensures that the generated reasoning translates into faithful and more accurate motion behaviors. In Fig. 11, we present two qualitative examples illustrating how post-training improves the model's motion fidelity. When the model reasons "decelerate, stop, and then accelerate at a stop sign," the aligned model produces

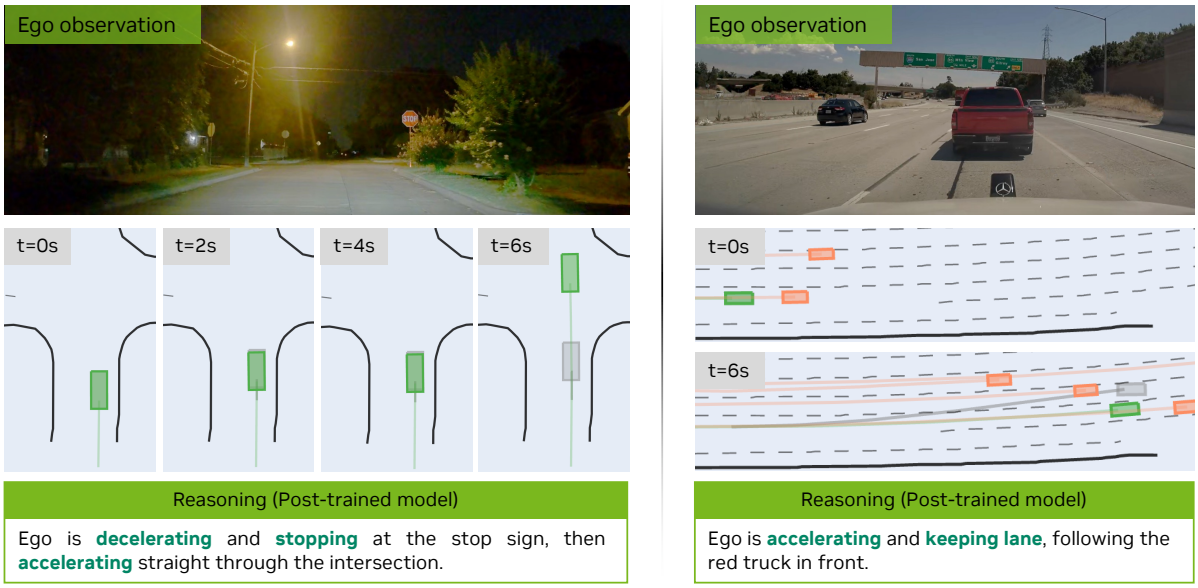

Figure 11: Post-training with the reasoning–action consistency reward improves motion fidelity. Grey motion denotes the most-likely rollout from the SFT-pretrained base model, and green motion denotes the most-likely rollout from the post-trained model. **Left**: The base model (grey) stops halfway and fails to resume motion, even though its reasoning trace correctly instructs the ego vehicle to accelerate after stopping. The post-trained model (green) executes the full causal sequence: decelerating, stopping, and accelerating once the intersection is clear. **Right**: When the reasoning instructs the ego vehicle to follow a lead vehicle, the post-trained model's generated motion maintains appropriate speed and lane position in accordance with its reasoning trace ("accelerating and keeping lane"), whereas the base model's generated motion changes the lane, drifting from the intended plan.

actions that faithfully follow this causal sequence (decelerating smoothly, coming to a complete stop, and accelerating only once the intersection is clear), whereas the SFT-pretrained model tends to stop halfway and never resume motion.

The Value of Imposing a Safety Reward. While reasoning and consistency rewards improve interpretability and causal grounding, they do not explicitly constrain the model to produce safe motion trajectories. To ensure physical safety, we introduce a safety reward that penalizes unsafe or physically implausible trajectories during post-training. Empirically, adding the safety reward further reduces the close encounter rate and stabilizes trajectory generation without compromising reasoning quality. As shown in Tab. 9, the full reward configuration achieves the lowest close encounter rate while maintaining improvements in ADE and reasoning–action consistency.

6.4. Ablation: VLM Backbone Selection

The choice of VLM backbone is critical for Alpamayo-R1's performance. In this section, we investigate two complementary aspects: the impact of model scale and the benefits of Physical-AI-focused pre-training. Together, these ablations demonstrate that both model capacity and domain-relevant pre-training are essential for strong driving performance.

6.4.1. Model Size Ablation

To investigate the impact of model capacity on driving performance, we first conduct baseline scaling experiments using general-purpose VLMs. Specifically, we evaluate three variants of our architecture with different backbone sizes: 0.5B, 3B, and 7B parameters. The 0.5B model uses a DINOv2 (Oquab et al., 2023) vision encoder combined with the Qwen2.5-0.5B (Qwen Team, 2024) language model, while the 3B and 7B models leverage Qwen2.5-VL-3B (Bai et al., 2025) and Qwen2.5-VL-7B (Bai et al., 2025), respectively. For this ablation study,

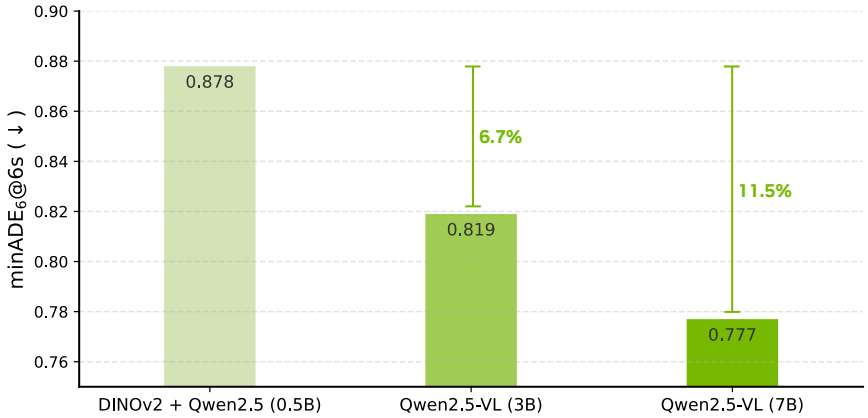

Figure 12: Impact of VLM backbone size on open-loop driving performance. All models are evaluated on $\mathcal{D}_{\text{overall}}$ with the same training data and hyperparameters.

all variants are trained on identical data with a reduced training budget compared to our main models, and evaluated on $\mathcal{D}_{\text{overall}}$ held-out test set without route information, using the minADE_6 metric over a 6 s horizon.

As shown in Fig. 12, we observe consistent improvements in open-loop performance as model size increases. The 7B model achieves a reduction of 11% in minADE_6 compared to the baseline of 0.5B, demonstrating that scaling the vision-language backbone enables better scene understanding and trajectory prediction. While these results confirm the importance of model capacity, they are based on general-purpose VLMs without domain-specific pre-training. As we demonstrate in the following subsection, incorporating Physical AI-focused pre-training (via Cosmos-Reason, Sec. 6.4.2) yields substantial further improvements, which is why our final Alpamayo-R1 models adopt Cosmos-Reason as their backbone.

6.4.2. Cosmos-Reason Physical AI Capabilities

While the scaling experiments above demonstrate the importance of model capacity, they do not address a critical question: given a fixed model size, does domain-specific pre-training matter? As described in Sec. 3, Alpamayo-R1 adopts Cosmos-Reason (NVIDIA et al., 2025) as its VLM backbone, specifically post-trained on Physical AI data including driving scenarios. To validate this architectural choice and demonstrate that Physical-AI-focused pre-training enhances driving-specific understanding beyond what scale alone provides, we evaluate Cosmos-Reason against comparable 7B-scale general-purpose VLMs on public driving benchmarks.

LingoQA Benchmark: Tab. 10 presents zero-shot evaluation results on the LingoQA benchmark (Marcu et al., 2024), which assesses vision-language models on driving scene understanding. Our Cosmos-Reason-7B model achieves 66.2% accuracy, outperforming various VLMs including GPT-4V (59.6%), Qwen2-VL-7B (52.6%), Qwen2.5-VL-7B (62.2%), InternVL3.5-8B (58.6%), and DeepSeek-VL-7B (46.4%). This improvement over the baselines demonstrates that Physical-AI-focused SFT significantly improves scene understanding capabilities for autonomous driving contexts, complementing the benefits of model scaling shown in Fig. 12.

Table 10: Zero-shot accuracy of various VLMs on the LingoQA benchmark (Marcu et al., 2024). Our Cosmos-Reason-7B model outperforms all baselines.

Model	GPT-4V	Qwen2-VL-7B	Qwen2.5-VL-7B	InternVL3.5-8B	DeepSeek-VL-7B	Ours
Lingo-Judge	59.6	52.6	62.2	58.6	46.4	**66.2**

These results confirm that *both* model capacity *and* domain-specific pre-training are essential for strong driving performance. This motivates our choice of Cosmos-Reason as the backbone for Alpamayo-R1, providing a strong foundation with Physical AI capabilities that general-purpose VLMs may otherwise not have.

Table 11: Comparison on trajectory decoding strategies. The models are trained and evaluated with route signals. The evaluation is on $\mathcal{D}_{\text{overall}}$ to show overall gains. Comfort (Accel) metric measures the percentage of predicted trajectories that are within a comfort range.

Strategy	**minADE$_6$@6s** $\downarrow$	**AlpaSim Score (at fault)** $\uparrow$	**Comfort (Accel)** $\uparrow$	**Rel. Decode Speed**$\uparrow$
Auto-Regressive	0.6811	0.59 ± 0.17	44.05%	$1.00\times$
Flow Matching	0.6440	1.27 ± 0.34	97.38%	$1.16\times$

Table 12: Relative comparison of different efficient vision encoding strategies on $\mathcal{D}_{\text{overall}}$.

Model	**Added Parameters** $\downarrow$	**Tokens per Image** $\downarrow$	**Rel. minADE$_6$** $\downarrow$
Baseline	0	160 ($1.0\times$)	0%
Triplane (Ivanovic et al., 2025)	6.3M	104 ($1.5\times$)	-3%
		45 ($3.6\times$)	$+4\%$
Flex (Yang et al., 2025)	61.6M	50 ($3.2\times$)	-3%
		32 ($5.0\times$)	-3%
		16 ($10\times$)	-2%
		8 ($20\times$)	-2%

6.5. Ablation: Action Modality Injection

We demonstrate the effectiveness of adopting a continuous action representation governed by unicycle dynamics with flow matching in Tab. 11. Specifically, we compare a baseline model trained to auto-regressively predict 6 discrete trajectory tokens against a model of identical size and training data that decodes trajectories via flow matching. The discrete trajectory tokenizer in the baseline auto-regressive model is pre-trained via VQGAN (Esser et al., 2021), which minimizes the number of output discrete tokens to reduce the auto-regressive decoding latency while maintaining low reconstruction error. During inference, we set $\delta_t = 0.2$, i.e., 5 steps, in flow matching to reduce latency with negligible performance degradation. As shown in Tab. 11, leveraging a dynamically governed continuous action space through flow-matching yields substantial improvements in both open-loop and closed-loop metrics, enhancing comfort and achieving faster inference speed.

6.6. Ablation: Efficient Vision Encoding

As discussed in Sec. 3.2.1, there are alternative methods for vision encoding that can be more efficient than the default single-image tokenizer in terms of tokens needed to represent multi-camera video inputs. To compare approaches, we choose a 4-camera setup, vary the vision encoder, and compare the resulting end-to-end model's open-loop driving quality via minADE$_6$ relative to the baseline.

As can be seen in Tab. 12, the triplane-based multi-camera tokenizer from Ivanovic et al. (2025) achieves nearly identical minADE$_6$ values as the baseline, while only adding 6.3M parameters and reducing sensor token counts by $3.6\times$. Flex (Yang et al., 2025) is able to achieve more drastic improvements, with a token compression of up to $20\times$ while only adding 61.6M parameters to the overall driving model and matching the driving quality of the baseline.

AR1 adopts single-image tokenization by default, as the optimal strategy can vary with the number of cameras, temporal frames, and camera resolutions. For example, a small number of cameras and short histories will favor single-image tokenization, more cameras and short histories will favor triplanes (Ivanovic et al., 2025), and more cameras and long history sequences will favor Flex (Yang et al., 2025).

Table 13: Inference runtime breakdown on an NVIDIA RTX 6000 Pro Blackwell. Alpamayo-R1 achieves real-time performance (99ms) by combining flow-matching-based trajectory decoding with efficient vision encoding.

Model Configuration	Vision Encoder	Prefilling	Reasoning Decoding	Trajectory Decoding	Total
Baseline (trajectory-only, flow matching)	3.43ms	16.54ms	–	8.75ms (5 steps)	29ms
Alpamayo-R1 (ours, flow matching)	3.43ms	16.54ms	70ms (40 tokens)	8.75ms (5 steps)	**99ms**
Alpamayo-R1 (auto-regressive traj)	3.43ms	16.54ms	70ms (40 tokens)	222ms (127 tokens)	312ms

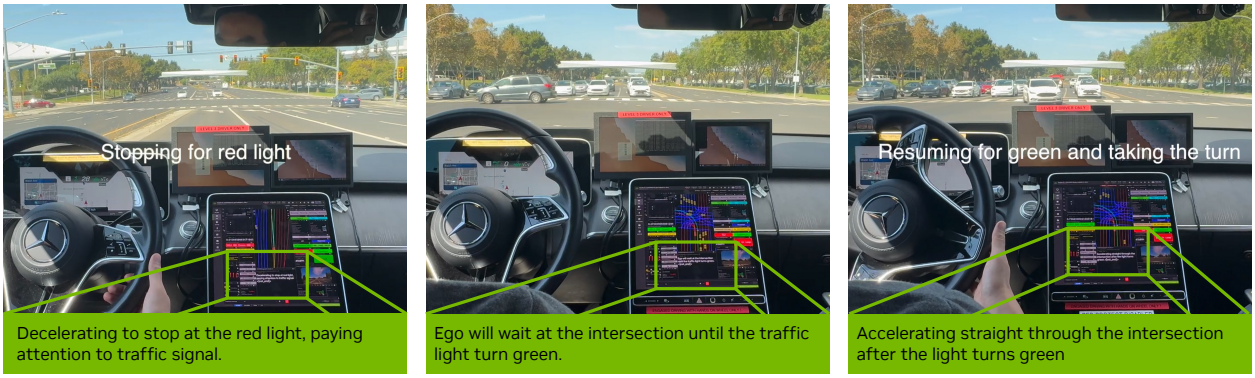

Figure 13: On-vehicle road test showing that AR1 generates reasoning trace in an intersection scenario. The ego vehicle first decelerates to stop due to the red light, then resumes when the light turns green and takes the turn.

6.7. On-Vehicle Road Tests

To validate the real-world deployment capability of AR1, we deployed the model in a test vehicle and conducted road testing in urban driving environments. The vehicle successfully navigated complex urban scenarios without human intervention, demonstrating the model's ability to handle real-world driving conditions beyond simulation. Fig. 13 shows an intersection where AR1 accurately identifies the traffic situation and produces clear and concise reasoning traces that lead to appropriate driving actions. These tests confirm that simulation improvements are transferred successfully to real-world autonomous driving scenarios.

Real-Time Inference Performance. A critical requirement for on-vehicle deployment is real-time inference capability. We benchmark AR1 on an NVIDIA RTX 6000 Pro Blackwell platform, achieving an end-to-end inference latency of 99ms, within the real-time requirements for autonomous driving (typically 100ms). Tab. 13 provides a detailed breakdown of the inference pipeline, comparing our approach against alternative design choices. The prefilling stage processes the visual tokens and route information through the transformer layers to generate the key-value cache, which is then used during both reasoning and trajectory decoding.

7. Conclusion

In this work, we present **Alpamayo-R1 (AR1)**, a vision-language-action model that integrates structured chain-of-thought reasoning capabilities with trajectory prediction to enhance autonomous driving performance, particularly in long-tail, safety-critical scenarios. To enable the model to generate causally-grounded reasoning, we introduce the **Chain of Causation (CoC)** dataset, constructed through a hybrid labeling pipeline that combines large-scale auto-labeling with humans in the loop. We further align reasoning with action through RL, ensuring that the generated reasoning traces are consistent with the executed driving behaviors. Our comprehensive evaluations across open-loop metrics, closed-loop simulation, and ablation studies demonstrate that AR1 achieves consistent improvements over end-to-end baselines, with particularly pronounced gains on challenging scenarios involving complex agent interactions.

Future Work. While our current evaluation focuses on internal datasets and the LingoQA benchmark, we plan to

extend our assessment to additional public benchmarks for autonomous driving planning and decision-making. This will provide a more comprehensive understanding of Alpamayo-R1's capabilities across diverse evaluation protocols and enable direct comparison with other state-of-the-art methods in the community. More broadly, several promising research directions remain open. First, *policy structuring*: while our flow-matching-based trajectory decoder provides kinematically feasible outputs, exploring hierarchical policy architectures that decompose high-level meta-actions into structured motion primitives could further improve interpretability and efficiency. Second, *reasoning on demand*: our current architecture generates reasoning traces for every input; future work could investigate adaptive mechanisms that selectively invoke reasoning only for safety-critical or ambiguous scenarios, enabling more efficient inference-time computation allocation similar to recent advances in test-time scaling (Yao et al., 2023; OpenAI, 2024); Third, *auxiliary task integration*: while AR1 focuses on trajectory prediction and causal reasoning, incorporating complementary self-supervised objectives, such as depth estimation, scene flow prediction, or 3D Gaussian Splatting representations, could improve the visual backbone's semantic understanding; Fourth, *world model integration*: our current approach predicts actions from observed states; incorporating learned world models could enable forward simulation and counterfactual reasoning, improving robustness in dynamic scenarios.

Open Source Release. We plan to release Alpamayo-R1 models on Hugging Face, along with a subset of the CoC dataset, augmenting the sensor data and labels available at NVIDIA's Hugging Face webpage, to advance research at the intersection of language-based reasoning and autonomous driving.

A. Contributors and Acknowledgments

A.1. Core Contributors

Yulong Cao, Tong Che, Yuxiao Chen, Wenhao Ding, Boris Ivanovic, Peter Karkus, Boyi Li, Tsung-Yi Lin, Patrick Langechuan Liu, Zhijian Liu, Jason Lu, Wenjie Luo, Marco Pavone, Ran Tian, Xinshuo Weng, Yan Wang, Tianjun Xiao, Xiaodong Yang, Yurong You, Xiaohui Zeng.

Data & Benchmarks: TX, XW, YC, WD, YW curated autonomous driving datasets and benchmarks.
Labeling Pipeline: XW, YC, WD, BL, XY, YW developed the reasoning trace labeling pipeline and the infrastructure.
Training Infrastructure: YY, WL, YW, WD built the supervised fine-tuning infrastructure; TC, RT, WL built the reinforcement learning infrastructure.
Vision Encoding: BI, YW developed the vision encoder.
Action Decoding: YY, YC built the flow-matching trajectory decoder.
Model Training: YY, WL, YW, WD, JL, ZL, PLL trained the VLA models with supervised fine-tuning; YW, WL, YY, XY, TL, XZ trained the Cosmos-Reason VLM backbone; RT, TC, YW, WL, YY, WD designed the post-training strategy and post-trained models with reinforcement learning; WD, YC designed the data mixture strategy.
Project Leads: YW, WL drove the project from concept to completion.
Program Architect and Project Manager: MP conceived, coordinated, and guided the overall effort. BI supported MP in coordination and guidance.

A.2. Contributors

Junjie Bai, Ke Chen, Jenna Diamond, Yifan Ding, Liang Feng, Jack Huang, Greg Heinrich, Pinyi Li, Dongran Liu, Ming-Yu Liu, Leo Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Yunfei Shi, Shida Shen, Sarah Tariq, Tilman Wekel, Eric Yang, Wenyuan Zhang.

Contributions. LP, JD led the human annotation effort. PM, GH, MR trained the vision encoder. ML provided Cosmos-Reason model support. YD processed cosmos AV data into training format. ZP improved the large-scale SFT training workflow. FL, JB provided support for the large-scale RL training infrastructure. ES curated and preprocessed driving data. KC, WZ, JH improved the CoC auto-labeling pipeline. SS developed the LLM-based evaluator for CoC reasoning traces. YS, EY, TW built the CoC labeling tools for human labeling. ST offered input and direction on the labeling pipeline. DL, PL and LM were instrumental in conducting the on-vehicle tests and model prefiling.

A.3. Acknowledgments

We thank Xinzhou Wu and Ali Kani for leadership and strategic support; Sachin Patil for general support in AV model training and deployment; Zhiding Yu, Guilin Liu, Max Li, Song Han, Hongxu Yin, Sifei Liu, and Yu-Wei Chao for valuable discussions on vision-language model training; Jesse Hong for running the CoC labeling pipeline; Richard Lin, Zi Wang, Walter Yu for improvements to the CoC auto-labeling pipeline; Anton Mitrokhin, Jacob Kern for improvements to the CoC human labeling pipeline; Martin Peng, Steve Hu, Andy Martin for dataset management and releases; Di Chen, Hanson Xu for help with model deployment; Chao Fang, Shuaijun Chen, and Niral Pathak for on-vehicle deployment support; Charles Vorbach, Zhenyi Zhang, Rachit Shah, Ritaank Tiwari for help with onboard vehicle deployment; Parixit Aghera, Ratin Kumar, Parag Mehendale, Niranjan Avadhanam, Rajath Shetty, Ronan LeToquin, Suraj Das and Ashley Hu for vehicle testing; Sachit Kadle, Annie Feng, and Zheng Lian for closed-loop simulation support; Maximilian Igl, Michael Watson, and Apoorva Sharma for closed-loop experimentation and metric implementations.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 5

[2] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. CoVLA: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1933–1943. IEEE, 2025. 5, 10

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 28

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. 20

[5] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005. 15

[6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End-to-End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016. 1

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 4

[8] 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*, 2025. 5, 10

[9] Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, et al. Language-image models with 3D understanding. In *International Conference on Learning Representations*, 2025. 3

[10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 2017. 3, 4, 19

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 5

[12] Charles Corbière, Simon Roburin, Syrielle Montariol, Antoine Bosselut, and Alexandre Alahi. Retrieval-based interleaved visual chain-of-thought in real-world driving scenarios. *arXiv preprint arXiv:2501.04671*, 2025. 4

[13] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pages 1268–1281. PMLR, 2023. 24

[14] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 3, 4, 6, 19, 20

[15] Xinpeng Ding, Jianhua Han, Hang Xu, Xiaodan Liang, Wei Zhang, and Xiaomeng Li. Holistic autonomous driving understanding by bird's-eye-view injected multi-modal large models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13668–13677, 2024. 5

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*. PMLR, 2017. 4

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 7

[18] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025. 2, 5, 9, 18

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 7, 30

[20] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *IEEE International Conference on Computer Vision*, 2021. 4

[21] Shiyu Fang, Yiming Cui, Haoyang Liang, Chen Lv, Peng Hang, and Jian Sun. CoReVLA: A dual-stage end-to-end autonomous driving framework for long-tail scenarios via collect-and-refine. *arXiv preprint arXiv:2509.15968*, 2025. 3

[22] Jiayuan Gu, Jiageng Chen, Yiming Liu, Hang Zhao, Yu Qiao, and Jifeng Dai. VAD: End-to-end video driving. In *IEEE/CVF International Conference on Computer Vision*, pages 5073–5083, 2023. 1

[23] Yuhan Hao, Zhengning Li, Lei Sun, Weilong Wang, Naixin Yi, Sheng Song, Caihong Qin, Mofan Zhou, Yifei Zhan, Peng Jia, et al. DriveAction: A benchmark for exploring human-like driving decisions in vla models. *arXiv preprint arXiv:2506.05667*, 2025. 4

[24] Deepti Hegde, Rajeev Yasarla, Hong Cai, Shizhong Han, Apratim Bhattacharyya, Shweta Mahajan, Litian Liu, Risheek Garrepalli, Vishal M Patel, and Fatih Porikli. Distilling multi-modal large language models for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27575–27585, 2025. 3

[25] Xinmeng Hou, Wuqi Wang, Long Yang, Hao Lin, Jinglun Feng, Haigen Min, and Xiangmo Zhao. DriveAgent: Multi-agent structured reasoning with LLM and multimodal sensor fusion for autonomous driving. *arXiv preprint arXiv:2505.02123*, 2025. 4

[26] Hanxue Hu, Ye Yuan, Hongyang Xu, Zhaoyang Chen, Ming Liang, Zhiding Li, Yuexin Ma, Xiaodong Shen, Yuning Chai, Xiaoqing Tan, et al. UniAD: Unified perception and prediction for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1

[27] 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. 2, 3

[28] Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, et al. DriveLMM-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. *arXiv preprint arXiv:2503.10621*, 2025. 5

[29] Boris Ivanovic, Cristiano Saltori, Yurong You, Yan Wang, Wenjie Luo, and Marco Pavone. Efficient multi-camera tokenization with triplanes for end-to-end driving. *IEEE Robotics and Automation Letters*, 10(11):11713–11720, 2025. 8, 30

[30] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pages 9902–9915, 2022. 18

[31] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024. 3

[32] Anqing Jiang, Yu Gao, Yiru Wang, Zhigang Sun, Shuo Wang, Yuwen Heng, Hao Sun, Shichen Tang, Lijuan Zhu, Jinhao Chai, et al. IRL-VLA: Training an vision-language-action policy via reward world model. *arXiv preprint arXiv:2508.06571*, 2025. 2, 3

[33] 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. 5

[34] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023. 18

[35] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[36] Samir Khaki, Junxian Guo, Jiaming Tang, Shang Yang, Yukang Chen, Konstantinos N. Plataniotis, Yao Lu, Song Han, and Zhijian Liu. SparseVILA: Decoupling Visual Sparsity for Efficient VLM Inference. In *ICCV*, 2025. 8

[37] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *European Conference on Computer Vision*, pages 563–578, 2018. 5

[38] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 22

[39] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. In *International Conference on Machine Learning*, 2023. 20

[40] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023. 4

[41] Sébastien Lefèvre, David Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1(1):1–14, 2014. 1

[42] Boyi Li, Ligeng Zhu, Ran Tian, Shuhan Tan, Yuxiao Chen, Yao Lu, Yin Cui, Sushant Veer, Max Ehrlich, Jonah Philion, et al. Wolf: Dense video captioning with a world summarization framework. *Transactions on Machine Learning Research*, 2025. 3

[43] Yiheng Li, Cunxin Fan, Chongjian Ge, Zhihao Zhao, Chenran Li, Chenfeng Xu, Huaxiu Yao, Masayoshi Tomizuka, Bolei Zhou, Chen Tang, et al. WOMD-Reasoning: A large-scale dataset for interaction reasoning in driving. *arXiv preprint arXiv:2407.04281*, 2024. 4

[44] 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*, 2025. 4

[45] Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-R1: Bridging reasoning and planning in VLMs for autonomous driving with reinforcement learning. *arXiv preprint arXiv:2506.18234*, 2025. 4

[46] Haicheng Liao, Hanlin Kong, Bonan Wang, Chengyue Wang, Wang Ye, Zhengbing He, Chengzhong Xu, and Zhenning Li. CoT-Drive: Efficient motion forecasting for autonomous driving with LLMs and chain-of-thought prompting. *IEEE Transactions on Artificial Intelligence*, 2025. 4

[47] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 2, 9, 18

[48] Wenru Liu, Pei Liu, and Jun Ma. DSDrive: Distilling large language model for lightweight end-to-end autonomous driving with unified reasoning and planning. *arXiv preprint arXiv:2505.05360*, 2025. 4

[49] Xueyi Liu, Zuodong Zhong, Yuxin Guo, Yun-Fu Liu, Zhiguo Su, Qichao Zhang, Junli Wang, Yinfeng Gao, Yupeng Zheng, Qiao Lin, et al. ReasonPlan: Unified scene prediction and decision reasoning for closed-loop autonomous driving. *arXiv preprint arXiv:2505.20024*, 2025. 4

[50] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7553–7560, 2023. 4

[51] Yuhang Lu, Jiadong Tu, Yuexin Ma, and Xinge Zhu. ReAL-AD: Towards human-like reasoning in end-to-end autonomous driving. *arXiv preprint arXiv:2507.12499*, 2025. 3

[52] 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. 2, 3

[53] Ziang Luo, Kangan Qian, Jiahua Wang, Yuechen Luo, Jinyu Miao, Zheng Fu, Yunlong Wang, Sicong Jiang, Zilin Huang, Yifei Hu, et al. MTRDrive: Memory-tool synergistic reasoning for robust autonomous driving in corner cases. *arXiv preprint arXiv:2509.20843*, 2025. 4

[54] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017. 9

[55] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. DRAMA: Joint risk localization and captioning in driving. *Winter Conference on Applications of Computer Vision*, 2023. 4, 10

[56] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. GPT-Driver: Learning to drive with GPT. *arXiv preprint arXiv:2310.01415*, 2023. 2, 3

[57] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. In *Conference on Language Modeling*, 2024. 2, 3

[58] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. LingoQA: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269, 2024. 5, 29

[59] Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety. *Advances in Neural Information Processing Systems*, 2024. 4

[60] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2Drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pages 292–308, 2024. 3, 5

[61] NVIDIA. Cosmos-RL: A flexible and scalable reinforcement learning framework. https://nvidia-cosmos.github.io/cosmos-rl/, 2025. 22

[62] NVIDIA, Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Liang Feng, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, Jacob Huffman, Jingyi Jin, Brendan Johnson, Rizwan Khan, George Kurian, Elena Lantz, Nayeon Lee, Zhaoshuo Li, Xuan Li, Maosheng Liao, Tsung-Yi Lin, Yen-Chen Lin, Ming-Yu Liu, Xiangyu Lu, Alice Luo, Andrew Mathau, Yun Ni, Lindsey Pavao, Wei Ping, David W. Romero, Misha Smelyanskiy, Shuran Song, Lyne Tchapmi, Andrew Z. Wang, Boxin Wang, Haoxiang Wang, Fangyin Wei, Jiashu Xu, Yao Xu, Dinghao Yang, Xiaodong Yang, Zhuolin Yang, Jingxu Zhang, Xiaohui Zeng, and Zhe Zhang. Cosmos-Reason1: From physical common sense to embodied reasoning, 2025. URL https://arxiv.org/abs/2503.15558. 5, 6, 20, 29

[63] NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez, Marco Pavone, Aaron Smith, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. AlpaSim: A modular, lightweight, and data-driven research simulator for end-to-end autonomous driving, 2025. URL https://github.com/NVlabs/alpasim. 23, 24, 26

[64] OpenAI. Learning to reason with LLMs, 2024. URL https://openai.com/index/learning-to-reason-with-llms/. 2, 4, 32

[65] OpenAI. GPT-5 system card. https://openai.com/index/gpt-5-system-card/, 2025. 15, 16

[66] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023. 28

[67] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016. 1

[68] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Association for Computational Linguistics*, pages 311–318, 2002. 15

[69] 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. 18

[70] Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, et al. AgentThink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. *arXiv preprint arXiv:2505.15298*, 2025. 4

[71] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. NuScenes-QA: A multi-modal visual question answering benchmark for autonomous driving scenario. In *AAAI Conference on Artificial Intelligence*, pages 4542–4550, 2024. 4

[72] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/. 28

[73] Qwen Team. Qwen3-VL: Sharper vision, deeper thought, broader action. https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afef&from=research.latest-advancements-list, 2025. 7

[74] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-only closed-loop autonomous driving with language-action alignment. In *IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025. 2, 3

[75] 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. 2, 3, 4

[76] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 4, 18, 19

[77] 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*, pages 256–274. Springer, 2024. 3, 5

[78] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 2015. 7

[79] Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*, 2024. 20

[80] Kexin Tian, Jingrui Mao, Yunlong Zhang, Jiwan Jiang, Yang Zhou, and Zhengzhong Tu. NuScenes-SpatialQA: A spatial understanding and reasoning benchmark for vision-language models in autonomous driving. *arXiv preprint arXiv:2504.03164*, 2025. 4

[81] Ran Tian and Kratarth Goel. Direct post-training preference alignment for multi-agent motion generation models using implicit feedback from pre-training demonstrations. *arXiv preprint arXiv:2503.20105*, 2025. 4, 19

[82] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *Conference on Robot Learning*, 2024. 3

[83] Ran Tian, Yilin Wu, Chenfeng Xu, Masayoshi Tomizuka, Jitendra Malik, and Andrea Bajcsy. Maximizing alignment with minimal feedback: Efficiently learning rewards for visuomotor robot policy alignment. *arXiv preprint arXiv:2412.04835*, 2024. 4, 19

[84] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 7

[85] Chris Urmson, John Anhalt, J. Andrew Bagnell, Christopher Baker, Robert Bittner, Michael N. Clark, John Dolan, Daniel Duggins, Todd Galatali, Christopher Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008. 1

[86] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017. 7

[87] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015. 15

[88] Feng Wang, Yaodong Yu, Guoyizhe Wei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Scaling laws in patchification: An image is worth 50,176 tokens and more. *arXiv preprint arXiv:2502.03738*, 2025. 7

[89] Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. DriveCoT: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024. 5

[90] 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. In *Advances in Neural Information Processing Systems*, 2022. 2, 3, 5, 18

[91] Maolin Wei, Wanzhou Liu, and Eshed Ohn-Bar. DriveQA: Passing the driving knowledge test. *arXiv preprint arXiv:2508.21824*, 2025. 4

[92] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized Architecture for Real-time Autonomous Driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024. 1

[93] Dongming Wu, Wencheng Han, Yingfei Liu, Tiancai Wang, Cheng-Zhong Xu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. *The Association for the Advancement of Artificial Intelligence*, 2025. 4

[94] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3DGUT: Enabling distorted cameras and secondary rays in gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26036–26046, 2025. 24

[95] Xinzhou Wu. Accelerate the future of AI-defined vehicles and autonomous driving, 2025. Available at https://www.nvidia.com/en-us/on-demand/session/gtc25-dd40000/. 1, 2, 5

[96] Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are VLMs ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives. *arXiv preprint arXiv:2501.04003*, 2025. 4

[97] Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M Wolff, and Xin Huang. VLM-AD: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*, 2024. 3

[98] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024. 5

[99] Jiawei Yang, Ziyu Chen, Yurong You, Yan Wang, Yiming Li, Yuxiao Chen, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. Towards efficient and effective multi-camera encoding for end-to-end driving. In *Under review*, 2025. 8, 30

[100] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, pages 11809–11822, 2023. 2, 3, 32

[101] Zhenlong Yuan, Jing Tang, Jinguo Luo, Rui Chen, Chengxuan Qian, 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. 2, 3

[102] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. FutureSightDrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025. 4

[103] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE International Conference on Computer Vision*, 2023. 7

[104] Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. ReWiND: Language-guided rewards teach robot policies without new demonstrations. In *Conference on Robot Learning*, 2025. 4, 19

[105] Xiangjun Zhang, Lin Qi, Qiwei Chen, Shuyang Su, Peng Liu, Zhiyuan Wang, Wei Zhang, and Daxin Zhao. Apollo EM Motion Planner. *arXiv preprint arXiv:1807.08048*, 2018. 1

[106] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *IEEE International Conference on Robotics and Automation*, pages 3560–3566, 2023. 18

[107] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025. 2, 3

[108] 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. 2, 3