# Extended Abstract

**Motivation:** Vision-language-action (VLA) models are promising for autonomous driving by mapping camera observations, ego state, and language context directly into future control trajectories, but autonomous driving places different demands than robot manipulation. A driving policy must produce smooth long-horizon actions, reason about road geometry, avoid unsafe trajectories, and replan fast enough for closed-loop control. Driving-specific VLA systems such as Alpamayo-R1 achieve strong open-loop accuracy, but they rely on multi-camera inputs and reasoning-heavy inference. In contrast, $\pi_{0.5}$ is a compact flow-matching VLA designed for real-time robot manipulation. It runs fast enough for control-rate deployment, but it has not been trained to drive. This creates the central question of our project: can a fast manipulation-trained flow policy be converted into a reliable driving policy?

**Method:** We introduce $\pi$-Drive, a driving version of $\pi_{0.5}$. Our method has two parts. First, we behavior-clone $\pi_{0.5}$ on real-world driving trajectories from NVIDIA PhysicalAI-AV. Each sample contains a front-camera frame, a low-dimensional ego state, a language navigation command, and a 6.4 second future egomotion trajectory. We represent actions as 64 timesteps of acceleration and curvature at 10 Hz, which are integrated through a unicycle kinematic model into an XYZ trajectory. This gives the policy a physically meaningful and kinematically feasible driving action space. The behavior cloning objective is the original flow-matching loss, but applied to driving action chunks instead of robot manipulation actions.

Second, we post-train the behavior-cloned policy with Flow-GRPO. For each scene, the policy samples a group of candidate trajectories. Each trajectory is scored with a composite driving reward combining drivable-area compliance, time-to-collision, comfort, route progress, command following, and reference-path alignment. Rewards are standardized within the group to produce group-relative advantages. Since $\pi_{0.5}$ does not produce autoregressive token log-probabilities, we reuse the flow-matching loss as a log-probability surrogate, $\log \pi_\theta(a|o) \approx -\mathcal{L}_{FM}(a, o)$. A KL penalty anchors the updated policy to the behavior-cloned reference.

**Preference Optimization Ablation:** We also study DPO as an ablation. Preference pairs are generated using Cosmos-3 as a VLM judge over rendered candidate trajectories. However, naive Flow-DPO is unstable because the flow-loss log-probability surrogate is unnormalized: the model can increase the preference margin by inflating the rejected trajectory's flow loss, corrupting the shared velocity field. Adding an imitation anchor stabilizes training, but the final policy still worsens Average Displacement Error (ADE) relative to behavior cloning. This suggests that VLM-generated safety preferences may optimize a different objective than open-loop human-trajectory matching.

**Results:** On 200 held-out PhysicalAI-AV clips with the same single front-camera input, $\pi$-Drive-GRPO achieves the best mean ADE and mean Final Displacement Error (FDE) among the single-camera models. Behavior cloning produces a non-degenerate driving policy, but it remains unstable on sharp turns and tends to over-damp longitudinal jerk. Flow-GRPO improves both trajectory accuracy and human-like smoothness, reducing mean ADE by approximately 13% relative to behavior cloning and outperforming a front-only Alpamayo-R1 baseline. The DPO ablation underperforms both Behavior Cloning and GRPO on ADE/FDE, reinforcing that not all post-training signals align with the evaluation objective.

**Discussion:** Our results suggest that much of the gap between compact VLAs and larger driving-specific VLAs comes from sensors and post-training rather than model size alone. At equal single-camera input, a compact flow policy can outperform a larger reasoning VLA while running substantially faster. Flow-GRPO is especially effective because it directly rewards driving-relevant properties that imitation alone does not enforce, such as lane compliance, progress, and comfort. However, our evaluation remains primarily open-loop. Future work should test whether these open-loop gains transfer to closed-loop driving on the golf cart and benchmark $\pi$-Drive on Bench2Drive with route completion, collision rate, and driving score.

**Conclusion:** $\pi$-Drive demonstrates that a manipulation-trained flow-matching VLA can be adapted into a real-time autonomous-driving policy through behavior cloning and reinforcement post-training. Behavior cloning teaches the model to drive; Flow-GRPO improves the policy beyond imitation; and DPO reveals the difficulty of aligning VLM preferences with trajectory metrics. This supports reinforcement post-training as a practical path for converting fast generalist VLAs into deployable driving policies.

# $\pi$-Drive: Reinforcement Post-Training Turns a Manipulation VLA into a Real-Time Driving Policy

**Felipe Barbosa**
Department of Computer Science
Stanford University
fbarbosa@stanford.edu

**Alex Kim**
Department of Computer Science
Stanford University
alexjkim@stanford.edu

**Mark Music**
Department of Computer Science
Stanford University
mmusic@stanford.edu

## Abstract

Vision-language-action (VLA) models offer a promising route toward autonomous driving: a single policy can map visual observations, ego state, and language context directly into future trajectories. However, driving requires smooth long-horizon control, safety-aware reasoning, and real-time replanning. Existing driving-specific VLAs achieve strong open-loop performance but often rely on multi-camera inputs and reasoning-heavy inference, making them difficult to deploy at closed-loop control rates. In this work, we introduce $\pi$-Drive, a driving-specialized version of $\pi_{0.5}$, a compact flow-matching VLA originally trained for robot manipulation. We first behavior-clone $\pi_{0.5}$ on large-scale real-world driving trajectories from NVIDIA PhysicalAI-AV, converting its action space from manipulation commands to 6.4 second acceleration-curvature trajectory chunks. We then improve the behavior-cloned policy using Flow-GRPO, a group-relative reinforcement learning method adapted to flow-matching policies through a flow-loss log-probability surrogate. On 200 held-out driving clips, Flow-GRPO reduces mean ADE by approximately 13% relative to behavior cloning and outperforms a front-camera-only Alpamayo-R1 baseline while running at roughly real-time control rate. We also study DPO with Cosmos-3-generated safety preferences and find that, although an imitation anchor stabilizes training, the resulting policy worsens open-loop ADE, suggesting that VLM safety preferences do not directly align with human-trajectory imitation. Our results show that reinforcement post-training can turn a fast manipulation VLA into a practical single-camera driving policy.

## 1 Introduction

Vision-language-action (VLA) models have recently emerged as a general framework for embodied control. Instead of separating perception, prediction, planning, and control into hand-engineered modules, a VLA policy can condition on visual observations, robot state, and language context to directly produce actions. This paradigm is attractive for autonomous driving, where a policy must interpret visual road structure, follow high-level intent, and generate future trajectories. However, driving also exposes the limits of current VLA systems. A deployed driving policy must generate smooth long-horizon actions, remain safe around other agents, and run at closed-loop control rates on edge hardware.

Stanford CS224R 2026 Final Report

Existing driving-specialized VLA models generally prioritize reasoning and accuracy. For example, NVIDIA's Alpamayo-R1 uses rich visual context and language reasoning to produce strong trajectory predictions, but its native setup depends on multiple cameras and slow inference (0.5 Hz on NVIDIA Jetson AGX Thor). This makes it difficult to use directly for real-time closed-loop control. In contrast, $\pi_{0.5}$ is a compact flow-matching VLA built for robot manipulation. It is fast enough for real-time deployment (10.6 Hz on NVIDIA Jetson AGX Thor), but was trained on basic robot manipulation tasks such as folding and placing objects rather than driving. This creates a natural tradeoff: driving-specific models are accurate but slow, while compact manipulation VLAs are fast but not immediately useful for autonomous driving.

In this work, we ask whether a fast manipulation-trained VLA can be converted into a reliable driving policy. We introduce $\pi$-Drive, a driving-specialized version of $\pi_{0.5}$ trained in two stages. First, we behavior-clone $\pi_{0.5}$ on real-world driving trajectories from NVIDIA PhysicalAI-AV. We replace the original manipulation action space with a driving action representation: a 6.4 second trajectory chunk represented as acceleration and curvature at 10 Hz. Second, we post-train the behavior-cloned policy with Flow-GRPO, a group-relative reinforcement learning method adapted to flow-matching policies.

The key technical challenge is that $\pi_{0.5}$ is not an autoregressive policy and does not expose token log-probabilities. Standard RL and preference optimization methods require some notion of policy likelihood. We address this by reusing the flow-matching loss as a log-probability surrogate, $\log \pi_\theta(a|o) \approx -\mathcal{L}_{FM}(a, o)$. This allows us to apply a PPO-style group-relative update to sampled trajectory chunks. Each group of trajectories is scored by a composite driving reward that combines drivable-area compliance, time-to-collision, comfort, progress, command following, and reference-path alignment.

Our experiments show that reinforcement post-training improves the behavior-cloned policy. On 200 held-out driving clips, $\pi$-Drive-GRPO reduced mean Average Displacement Error (ADE) by approximately 13% relative to behavior cloning and outperforms a front-camera Alpamayo-R1 baseline. We also evaluate a DPO ablation using Cosmos-3-generated safety preferences. Although adding an imitation anchor stabilizes DPO training, the resulting policy worsened ADE, suggesting that VLM preferences over safety do not necessarily align with open-loop human-trajectory planning.

Our contributions are:

1. We adapt $\pi_{0.5}$, a manipulation-trained flow-matching VLA, into a single-camera autonomous-driving policy.
2. We introduce a behavior cloning pipeline that trains the model to output 6.4 second acceleration-curvature trajectory chunks.
3. We adapt Flow-GRPO to flow-matching policies using a flow-loss log-probability surrogate and a composite driving reward.
4. We empirically show that Flow-GRPO improves open-loop trajectory accuracy and human-like smoothness, while DPO with VLM safety preferences does not improve ADE.

## 2 Related Work

### 2.1 Vision-language-action models

VLA models extend vision-language models (VLM) into embodied control by conditioning action generation on visual observations, robot state, and language instructions. Rather than producing only text, these models produce continuous or discrete actions that can be executed by an embodied agent. Models such as $\pi_{0.5}$ are especially relevant because they combine a pretrained vision-language backbone with a continuous action expert, allowing the policy to use visual and semantic context while still producing low-level control outputs (Black et al. (2025a)). However, most generalist VLA work has focused on robot manipulation, where actions are short-horizon end-effector commands and the environment is relatively local. Driving has different temporal and geometric structure: the policy must output smooth multi-second trajectories, obey kinematic constraints, remain inside the drivable corridor, and replan at control-rate frequencies. Our work studies whether the visual grounding and action-generation capabilities of a manipulation-trained VLA can be transferred into autonomous driving through behavior cloning and reinforcement post-training.

### 2.2 Alpamayo-R1

Autonomous-driving trajectory prediction has typically relied on driving-specific architectures, multi-camera sensor suites, map information, agent histories, or explicit planning modules to predict future ego motion. Recent VLA-style driving models, such as Alpamayo-R1, combine visual-language reasoning with action prediction and achieve strong open-loop trajectory performance in complex scenes (Wang et al. (2025)). However, these systems often obtain their strongest results using richer sensor inputs and slower inference. This makes it difficult to separate the contribution of model architecture from the contribution of the input sensor suite. In our evaluation, Alpamayo-R1 is important because it represents the opposite tradeoff from $\pi_{0.5}$: it is driving-specialized and reasoning-heavy, while $\pi_{0.5}$ is compact and fast but not originally trained to drive. To isolate model quality from sensor advantage, we compare against a front-camera-only Alpamayo-R1 baseline. This gives both systems the same visual input and tests whether a compact single-camera flow policy can be competitive under matched sensing constraints.

This tradeoff is architectural, not just model size: Alpamayo-R1's autoregressive reasoning pass dominates its latency, whereas $\pi_{0.5}$ decodes the full action chunk in a fixed number of parallel flow steps (Black et al. (2024); Driess et al. (2025); Black et al. (2025b)). We return to this speed gap, and show it comes at no accuracy cost, in Section 6.1.

### 2.3 Behavior Cloning

Behavior cloning is a standard imitation-learning approach for autonomous driving: given an observation the policy is trained to predict the future trajectory demonstrated by an expert driver. In our setting, behavior cloning is necessary because the original $\pi_{0.5}$ policy was trained for manipulation, not driving. We therefore replace its manipulation action space with a driving action space and fine-tune it on real-world trajectories from NVIDIA PhysicalAI-AV (NVIDIA (2025)). Each output is a 128-dimensional action chunk, corresponding to 64 timesteps of acceleration and curvature over a 6.4 second horizon. These controls are integrated through a unicycle kinematic model, which makes the generated trajectory dynamically meaningful rather than an unconstrained set of waypoints. Behavior cloning gives the model a useful prior, but it is limited by the supervised objective: it only maximizes likelihood of logged human trajectories and doesn't directly optimize safety, drivable-area compliance, progress, or comfort. This motivates reinforcement post-training as a second stage (Karkus et al. (2025)).

Flow matching trains a generative model by learning a continuous velocity field that transports noise to data. In $\pi_{0.5}$, the action expert is a conditional flow-matching generator. Given an observation $o$, a ground-truth action chunk $a$, Gaussian noise $\epsilon$, and flow time $\tau$, the model observes an interpolated action

$$a_\tau = \tau a + (1 - \tau)\epsilon$$

and learns to predict the target velocity

$$u = a - \epsilon.$$

The training objective is a mean-squared error between the predicted velocity field and this target velocity. This parametrization is well suited to continuous control because the model can generate an entire high-dimensional action chunk in parallel instead of autoregressively predicting one action at a time. In $\pi$-Drive, this action chunk corresponds to the future acceleration-curvature sequence. A central technical challenge is that flow policies do not expose normalized autoregressive token log-probabilities. This matters because RL and preference-optimization methods like GRPO and DPO are normally formulated in terms of policy likelihood ratios. Our method therefore reuses the negative flow-matching loss as a surrogate log-probability,

$$\log \pi_\theta(a|o) \approx -\mathcal{L}_{FM}(a, o),$$

which allows post-training objectives to be applied to a flow-matching action policy.

### 2.4 Reinforcement Learning Post-training

Reinforcement learning post-training can improve generative policies beyond imitation by optimizing task-specific rewards. This is particularly useful for driving because many desirable properties are easier to express as reward terms than as supervised targets. For example, lane compliance,

time-to-collision, route progress, command following, and comfort are not directly optimized by a pure behavior-cloning loss. We adapt Flow-GRPO to the driving setting (Liu et al. (2025)). For each scene, the policy samples a group of candidate trajectories. Each candidate is integrated into physical space and scored using a composite driving reward. Rewards are then normalized within the group to produce group-relative advantages,

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}.$$

This avoids training a separate critic and uses the group mean as a baseline. The policy is updated toward higher-reward trajectories using a PPO-style clipped objective, while a KL penalty anchors the policy to the behavior-cloned reference. The main adaptation in our work is making this update compatible with a flow policy by computing likelihood ratios through the flow-loss surrogate rather than log-probabilities.

Direct Preference Optimization (DPO) trains a policy from preferred and rejected samples without explicitly fitting a reward model (Rafailov et al. (2023)). Diffusion-DPO extends this idea to diffusion-style generative models, making it relevant for flow-based action policies (Wallace et al. (2023)). We evaluate a DPO-style ablation using Cosmos-3 as a VLM judge over candidate driving trajectories (NVIDIA (2026a)). This ablation tests whether VLM safety judgments can provide a useful post-training signal beyond hand-designed rewards.

# 3 Method

$\pi$-Drive is built in two stages: behavior cloning on real-world driving trajectories to obtain a non-degenerate policy ($\pi$-Drive-BC), followed by group-relative reinforcement post-training.

## 3.1 Behavior Cloning

### 3.1.1 Architecture and Action Space

We initialize from the public $\pi_{0.5}$ checkpoint (`pi05_base`): a PaliGemma backbone (SigLIP vision + Gemma-2B) feeding a Gemma-300M flow-matching action expert, 3.35B parameters total. We apply LoRA (rank 32, $\alpha = 64$) to the Gemma-2B backbone but fully fine-tune the action expert, since the manipulation-trained action representation does not transfer to driving; the action projection layers are reshaped (32 to 128 dimensions) and reinitialized. We replace $\pi_{0.5}$'s manipulation action space with a kinematic driving representation: a 6.4 s trajectory as 64 steps of (acceleration, curvature) at 10 Hz, integrated through a unicycle model (Alpamayo's `UnicycleAccelCurvatureActionSpace`) so every sampled trajectory is kinematically feasible by construction.

### 3.1.2 Behavior Cloning Objective

We train $\pi$-Drive-BC with $\pi_{0.5}$'s conditional flow-matching loss applied to driving chunks. Given an observation $o$ (front camera, ego state, navigation command) and a ground-truth action chunk $a$, we sample $\tau \in [0, 1]$ and $\epsilon \sim \mathcal{N}(0, I)$, form $a^\tau = \tau a + (1 - \tau)\epsilon$, and regress the conditional velocity field:

$$\mathcal{L}_{FM} = \mathbb{E}_{\tau,\epsilon,(o,a)} \left\| v_\theta(a^\tau, \tau, o) - (a - \epsilon) \right\|^2. \tag{1}$$

We train for 15k steps on 8×H100 GPUs in roughly 6 hours, producing $\pi$-Drive-BC, our imitation baseline and the frozen reference policy for post-training. Trained purely with this objective, however, the policy begins to overfit to the NVIDIA PhysicalAI-AV distribution: as shown in Figure 1, on the log scale the training loss keeps decreasing while the evaluation loss plateaus near $0.1$ after roughly 6k steps, widening the train–eval gap. This onset of overfitting under pure imitation is a key motivation for improving the policy through reinforcement post-training rather than longer behavior cloning.

### 3.1.3 Navigation Conditioning

Each sample is conditioned on a discrete navigation command, which cannot be recovered by a deterministic function of the future trajectory: the same lateral displacement arises both from following a curving road and from a deliberate turn at a junction, so a geometric threshold would label every bend as a commanded turn. The correct label is the high-level intent a route planner issues

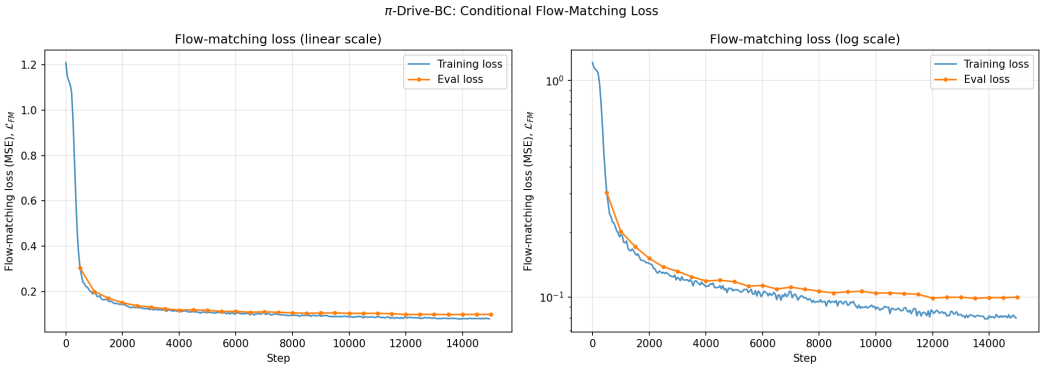


Figure 1: Behavior cloning conditional flow-matching loss on NVIDIA PhysicalAI-AV (linear scale, left; log scale, right). On the log scale, the training loss continues to decrease while the evaluation loss stagnates near 0.1 after roughly 6k steps, indicating the onset of overfitting under pure behavior cloning.

at deployment (continue, turn at the junction, change lanes), which depends on scene semantics such as whether an intersection is present. We therefore generate labels with Gemini 2.5 Flash, prompting it over the front-camera view and realized trajectory to classify the maneuver intent, a judgment a closed-form rule cannot make.

Because the command strongly predicts the maneuver, an always-conditioned policy leans on it and ignores the camera. Prompt dropout is the only mechanism preventing mode collapse here: without it the policy reproduces the label-dictated trajectory and collapses the full range of visually-grounded behaviors onto a single intent-driven mode. We therefore replace the command with a generic `"drive"` token on 30% of samples, forcing the policy to drive from vision alone. This classifier-free guidance lets the command act as a hint that disambiguates intersections rather than a crutch.

### 3.2 Post-training: Flow-GRPO and DPO

Figure 1 shows behavior cloning has hit its ceiling: the evaluation loss stagnates while the training loss keeps falling, so more imitation only overfits. This is fundamental, not a tuning issue. Behavior Cloning (BC) can at best match the demonstrator and inherits its failure modes, the sharp-turn instability (Figure 2) and over-damped, timid jerk (Table 2); it cannot prefer a trajectory better than the one it was shown. Reinforcement post-training lifts this ceiling by optimizing driving-relevant rewards directly, recovering committed, human-like driving that more behavior cloning could not.

#### 3.2.1 Flow-GRPO

For each observation $o$, the behavior-cloned policy samples a group of 8 candidate action chunks,

$$\{a_i\}_{i=1}^{8} \sim \pi_\theta(\cdot \mid o),$$

where each $a_i$ is a 64-step sequence of acceleration and curvature. Each action chunk is integrated through the same unicycle dynamics used during behavior cloning to obtain a physical ego trajectory. We then score each candidate with a composite driving reward,

$$R_i = 0.5R_{\text{drive},i} + 0.3R_{\text{cmd},i} + 0.2R_{\text{ref},i}.$$

The driving term follows a PDMS-style structure:

$$R_{\text{drive},i} = \text{DAC}_i \cdot \frac{5\,\text{TTC}_i + 2\,\text{comfort}_i + 5\,\text{progress}_i}{12}.$$

Here, DAC is drivable-area compliance, measured as the fraction of trajectory waypoints that remain inside the drivable corridor. We use it as a multiplicative gate, so trajectories that leave the lane sharply reduce the entire driving-quality term. TTC measures time-to-collision safety against the lead vehicle, comfort penalizes aggressive acceleration and jerk, and progress rewards forward route progress. The command term $R_{\text{cmd}}$ scores whether the trajectory's inferred meta-action matches the language navigation command. The reference term is a clipped ADE guardrail,

$$R_{\text{ref},i} = \max(-\text{ADE}(a_i, a_{\text{GT}}), -3.0),$$

which encourages proximity to the human trajectory but saturates once the candidate is more than 3 meters away. This makes reference matching a stabilizing guardrail rather than the dominant objective.

Rather than training a critic, we normalize rewards within the sampled group:

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon},$$

This produces a group-relative advantage where trajectories are reinforced only if they are better than the other samples from the same scene. This is important for driving because absolute reward magnitudes vary substantially across scenes, especially when sharp turns, straight highways, and intersections may all have different reward ranges.

The main technical issue is that $\pi_{0.5}$ is a flow-matching policy, not an autoregressive policy, so it does not expose normalized action log-probabilities. We therefore use the negative conditional flow-matching loss as a surrogate log-probability:

$$\log \pi_\theta(a_i \mid o) \approx -\mathcal{L}^\theta_{FM}(a_i, o).$$

This gives the likelihood ratio

$$r_i(\theta) = \exp\left(-\mathcal{L}^\theta_{FM}(a_i, o) + \mathcal{L}^{\theta_{\text{old}}}_{FM}(a_i, o)\right).$$

We optimize a PPO-style clipped objective,

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_i\left[\min\left(r_i(\theta)A_i, \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)A_i\right)\right] + \beta\, D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{BC}}),$$

where $\pi_{\text{BC}}$ is the frozen behavior-cloned reference policy. The KL term prevents the policy from moving too far from the imitation prior, while the group-relative reward pushes the model toward trajectories that are safer, smoother, and more committed than the average rollout.

### 3.2.2 DPO Ablation

We also evaluate a preference-optimization alternative based on Cosmos-3. This choice is natural because Cosmos is the spatially-aware backbone used in Alpamayo's reasoning stack, so it provides a model-family-aligned judge for driving scenes. We expected Cosmos-3 to improve over Alpamayo's original Cosmos-Reason backbone: as a newer and stronger VLM, it should, in principle, provide better scene understanding, better trajectory-level safety judgements, and more reliable preference labels. For each scene, we render candidate trajectories and ask Cosmos-3 to select a preferred trajectory $a^+$ and a rejected trajectory $a^-$. We then apply a DPO-style loss using the same flow-loss surrogate:

$$\mathcal{L}_{\text{DPO}} = -\log\sigma\left(\beta\left[\Delta_\theta(a^+, a^-) - \Delta_{\text{ref}}(a^+, a^-)\right]\right),$$

where

$$\Delta_\theta(a^+, a^-) = \log\pi_\theta(a^+|o) - \log\pi_\theta(a^-|o) \approx -\mathcal{L}^\theta_{FM}(a^+, o) + \mathcal{L}^\theta_{FM}(a^-, o).$$

and $\beta$ is the DPO inverse-temperature parameter controlling the strength of the preference update. The larger $\beta$ is, the more aggressively it will increase the margin between preferred and rejected trajectories.

However, this preference signal did not translate into better open-loop trajectory prediction. Because the flow-loss surrogate is not a normalized likelihood, naive Flow-DPO was unstable: the model can increase the preference margin by increasing the flow loss on the rejected trajectory rather than improving the preferred one. This corrupted the shared velocity field and degraded trajectory accuracy. We therefore added an imitation anchor,

$$\mathcal{L} = \mathcal{L}_{\text{DPO}} + \lambda\mathcal{L}_{FM}(a_{\text{GT}}, o),$$

which keeps the learned velocity field close to the human trajectory distribution. This stabilizes optimization, but the final DPO policy still underperforms behavior cloning on ADE and FDE (Final Displacement Error). The result suggests that a stronger Cosmos-family VLM judge may optimize a different objective from open-loop human-trajectory matching: a trajectory can appear safer or more reasonable to the VLM while moving farther from the demonstrated path.

# 4 Experimental Setup

We evaluate all models on a fixed set of 200 held-out clips from NVIDIA PhysicalAI-AV. Each clip contains a single front-camera image, ego state, a navigation command, and a 6.4 s ground-truth future ego trajectory. To isolate model quality from sensory advantage, all models are evaluated with the same single front-camera input.

We compare four models: Alpamayo-R1, $\pi_{0.5}$ behavior cloning ($\pi$-Drive-BC), $\pi_{0.5}$ post-trained with Flow-GRPO, and the Cosmos-DPO ablation. Each model produces one sampled trajectory per clip. We report open-loop displacement error using ADE and FDE, both in meters. We also report longitudinal jerk ratios relative to the human ground-truth trajectory. A jerk ratio of 1.0 indicates human-like smoothness, while values below 1.0 indicate smoother and more timid motion.

# 5 Results

We evaluate whether a manipulation-trained flow-matching policy can be adapted into a competitive single-camera driving policy, and whether group-relative post-training improves it over behavior cloning. All models are evaluated on the same single front-camera feed to isolate model quality from sensor advantage. We therefore run Alpamayo-R1 front-only, matching the input given to $\pi_{0.5}$ rather than its native four-camera setup, so any difference reflects the model rather than the sensor suite. Results are reported on a fixed set of 200 held-out PhysicalAI-AV clips with shared ground truth and a single sampled rollout per model.

## 5.1 Quantitative Analysis

Table 1 reports open-loop accuracy across the four single-camera models. Flow-GRPO is the strongest model overall, achieving the lowest mean ADE (3.58 m), median ADE (2.95 m), and mean FDE (10.19 m). It improves over the behavior-cloned baseline ($\pi_{0.5}$ BC) on three of four accuracy metrics and, notably, matches or beats front-only Alpamayo-R1 at equal input despite being roughly $3\times$ smaller and single-shot, with no reasoning-time search. The behavior-cloned policy is competitive in its own right, sitting close to front-only Alpamayo while remaining far more efficient.

The Cosmos-DPO ablation is the weakest model on accuracy, regressing below BC on all four metrics. This is consistent with its objective: the VLM-judge preferences optimize scene-level safety rather than proximity to the ground-truth trajectory, which is the quantity ADE and FDE measure.

| Model | Input | m.ADE | md.ADE | m.FDE | md.FDE |
|---|---|---|---|---|---|
| Alpamayo-R1 (front) | hist | 4.23 | **2.85** | 12.40 | 8.75 |
| $\pi_{0.5}$ GRPO | state | **3.58** | 2.95 | **10.19** | 8.63 |
| $\pi_{0.5}$ BC | state | 4.13 | 3.48 | 11.04 | **8.60** |
| $\pi_{0.5}$ Cosmos-DPO | state | 4.69 | 4.02 | 12.31 | 10.63 |

Table 1: Open-loop accuracy on 200 fixed held-out clips with identical single-camera input. ADE and FDE are in meters; lower is better. Best values are bolded. Flow-GRPO is the strongest single-camera model and beats front-only Alpamayo-R1 at equal input.

Beyond displacement error, we measure ride smoothness through longitudinal jerk, reported in Table 2 as the ratio of model jerk to human ground-truth jerk. A ratio of 1.0 is human-like; values below 1.0 indicate motion that is smoother and more timid than a human driver. All models fall below 1.0, but Flow-GRPO is closest to human on all four ratios (RMS and peak, mean and median), indicating it under-reacts the least. Behavior cloning over-damps more heavily (ratios near 0.5), and front-only Alpamayo and Cosmos-DPO are the most timid (ratios near 0.3–0.4).

| Model | RMS mean | RMS median | Peak mean | Peak median |
|---|---|---|---|---|
| Alpamayo-R1 (front) | 0.38 | 0.32 | 0.40 | 0.29 |
| $\pi_{0.5}$ GRPO | **0.71** | **0.60** | **0.63** | **0.50** |
| $\pi_{0.5}$ BC | 0.52 | 0.48 | 0.44 | 0.39 |
| $\pi_{0.5}$ Cosmos-DPO | 0.38 | 0.32 | 0.43 | 0.27 |

Table 2: Longitudinal jerk versus human ground truth, reported as the model-to-human ratio. A value of 1.0 is human-like and values below 1.0 are smoother. Flow-GRPO is closest to human on all four measures; best values are bolded.

### 5.2 Qualitative Analysis

Figure 2 compares predicted trajectories against ground truth on two representative turning clips, one a night right turn and one a daytime left turn, both at highway speed. The qualitative behavior tracks the quantitative results and exposes a distinct failure signature for each model.

#### 5.2.1 $\pi_{0.5}$ BC

The behavior-cloned model learns the overall trajectory shape and matches front-only Alpamayo on accuracy, but it is unstable on sharp turns, occasionally over-turning and drifting outside the lane. Its jerk ratios near 0.5 indicate a smooth but timid driver that under-commits to corrections. These turn failures are exactly the behavior post-training is designed to fix.

#### 5.2.2 $\pi_{0.5}$ GRPO

Flow-GRPO tracks the ground-truth path most closely on both turns and removes the over-turning instability seen in BC. Because it samples eight candidates per scene and reinforces those with higher group-relative advantage, it learns to commit to the correct turn rather than hedging. It is also closest to human jerk, indicating that the reward signal recovers committed, human-like driving behavior that pure imitation leaves on the table.

#### 5.2.3 $\pi_{0.5}$ Cosmos-DPO

The DPO ablation drifts away from the ground-truth path, in one case turning the wrong direction entirely. It underperforms both BC and GRPO on ADE and FDE. The VLM-judge preferences reward scene-level safety, which aligns only weakly with proximity to the human trajectory, so optimizing them moves the policy in the wrong direction for displacement error. Stronger imitation anchoring and trajectory-aware preferences would be needed for DPO to help.

#### 5.2.4 Front-only Alpamayo-R1

The four-camera baseline, restricted to a single camera to match our input, systematically under-turns and flattens its predictions toward a near-straight path. This is the expected signature of out-of-distribution input: a model trained on a four-camera rig loses the lateral context when given only the front view.

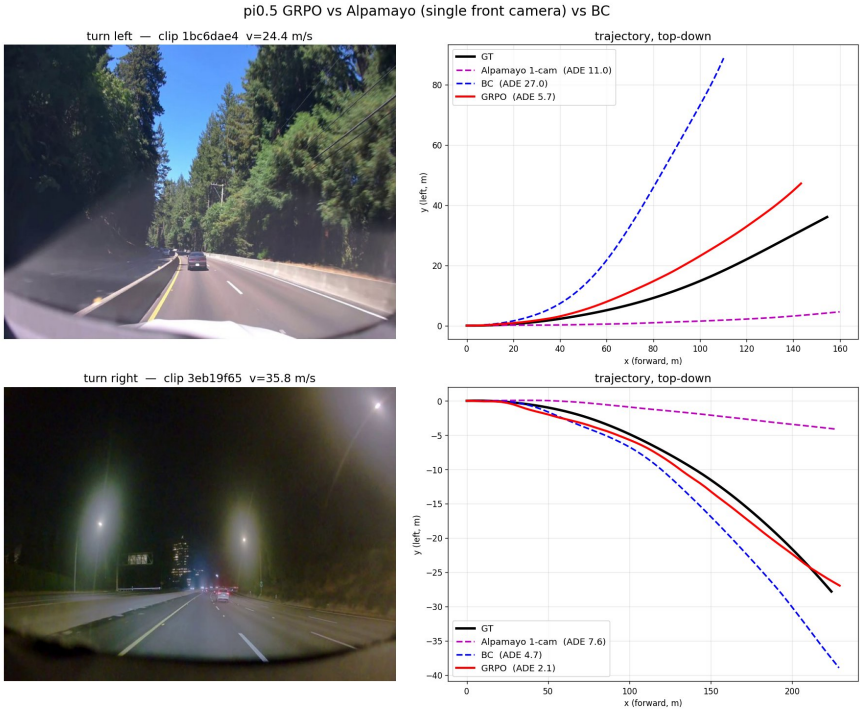


Figure 2: Top-down predicted paths versus ground truth (GT) on a night right turn (top) and a daytime left turn (bottom), single front camera; per-clip ADE is shown in the legend. GRPO tracks GT most closely, BC over-turns, the Cosmos-DPO ablation drifts the wrong way, and front-only Alpamayo under-turns.

# 6 Discussion

Our results show that a compact flow-matching policy built for manipulation can be turned into a competitive, real-time single-camera driving policy, and that group-relative post-training is the component that closes the gap to a much larger reasoning VLA. In this section we analyze the behavior, mechanisms, and limitations of each training stage.

## 6.1 Sensors Versus Model Scale

Under input parity, a $\sim$3B flow policy rivals a 10B reasoning VLA while running roughly $20\times$ faster on the same edge GPU (10.7 Hz vs 0.5 Hz). This speedup also uses a TensorRT-optimized implementation of $\pi_{0.5}$, but the main advantage is architectural rather than only systems-level: Alpamayo-R1 autoregressively decodes a chain-of-thought before the trajectory, so its latency scales with the reasoning trace, whereas $\pi_{0.5}$ decodes the full 6.4 s chunk in a fixed number of parallel flow steps and uses a continuous action expert that preserves fast inference (Black et al. (2024); Driess et al. (2025); Black et al. (2025b); NVIDIA (2026b)). Crucially, this speed costs nothing in accuracy: at equal single-camera input, $\pi$-Drive matches or beats front-only Alpamayo-R1 on ADE and FDE (Table 1), so the gap localizes to the sensor suite rather than model capacity. Front-only Alpamayo's degradation is itself informative: it systematically under-turns and flattens toward a near-straight path, the expected signature of a model trained on a four-camera rig losing lateral context. This is what makes on-vehicle deployment practical: control-rate inference on a single edge GPU, without multi-camera rigs or multi-GPU serving.

## 6.2 Why Post-Training Helps

Flow-GRPO improves over behavior cloning on almost every accuracy metric and, more importantly, removes BC's sharp-turn instability where the policy veers out of lane. The enabling mechanism is specific to flow policies: because $\pi_{0.5}$ exposes no token log-probabilities, we reuse its flow-matching loss as a log-probability surrogate ($\log \pi \approx -\mathcal{L}_{\text{FM}}$), which makes a group-relative objective possible without training a separate critic. Sampling eight candidates per scene and reinforcing those with

higher group-relative advantage teaches the policy to commit to the correct turn rather than hedge. The jerk results corroborate this: GRPO is closest to human on all four ratios, indicating it under-reacts the least, while BC over-damps toward a timid profile. That a lightweight objective recovers committed, human-like turning shows the reward signal captures behavior a strong demonstrator alone does not.

### 6.3 Why Preference Optimization Fails

The Cosmos-DPO ablation is a cautionary result. It regresses below BC on every accuracy metric because its VLM-judge preferences optimize scene-level safety rather than proximity to the human trajectory, which is the quantity ADE and FDE measure. The instability also has a concrete mechanical cause: because the flow-matching surrogate is unnormalized, naive Flow-DPO widens the preference margin by inflating the rejected action's flow loss, which corrupts the shared velocity field. An imitation anchor stabilizes training but does not change the fact that the objective is misaligned with trajectory accuracy. The lesson is that the choice of training signal matters as much as the optimization method.

### 6.4 Limitations

Our main evaluation is open-loop, and displacement error understates closed-loop drivability, the very property Flow-GRPO is rewarded for. The qualitative comparison covers a small number of clips, so the per-model failure signatures are illustrative rather than exhaustive. Finally, our preference-optimization signal comes from a VLM safety judge rather than a closed-loop driving outcome, which we have shown to be the wrong lever for the metrics we report.

## 7 Conclusion

We asked whether a fast, manipulation-trained flow-matching VLA could be turned into a reliable driving policy. The answer is yes. With behavior cloning and group-relative post-training, $\pi$-Drive matches a far larger reasoning VLA on open-loop accuracy at equal single-camera input while running at control rate on a single edge GPU. To our knowledge this is the first adaptation of a flow-matching manipulation policy to autonomous driving, and the first application of group-relative RL to a flow-matching action head in this domain.

The contribution is less a single number than a recipe: a compact policy paired with a reward aligned to the right objective can stand in for model scale and sensor count in single-camera driving. This reframes where effort should go. Rather than larger backbones or richer sensor rigs, the leverage is in data and post-training, applied to a model small enough to deploy. We see $\pi$-Drive as a step toward driving policies that are fast enough to deploy on a vehicle and good enough to trust.

**Limitations and Future Directions:** Our evaluation is open-loop and displacement error understates the closed-loop drivability Flow-GRPO is rewarded for. The decisive next test is closed-loop: rolling $\pi$-Drive out on our instrumented golf cart and benchmarking it on Bench2Drive with route completion, collision rate, and driving score. We also plan to revisit preference optimization with trajectory-aware, closed-loop safety signals rather than static VLM judgments, which we showed to be misaligned with the metrics that matter.

**Broader Impact:** Our findings point toward a practical path for deploying capable driving policies on resource-constrained, single-sensor platforms, removing the multi-camera rigs and multi-GPU inference that reasoning-heavy VLAs require. By showing that a small, fast manipulation policy can be post-trained into a control-rate driving policy on a single edge GPU, we lower the barrier to on-vehicle deployment, while retaining the flexibility to escalate to richer sensors or larger backbones where the application demands it.

## 8 Team Contributions

- **Felipe Barbosa:** Led all post-training, including the Flow-GRPO implementation, composite reward design, group-relative advantage computation, DPO ablations, Cosmos-3 preference pipeline, and Modal RL training jobs.

- **Alex Kim:** Led all evaluation work, including the open-loop evaluation harness, ADE/FDE computation, longitudinal jerk analysis, qualitative trajectory visualizations, Alpamayo-R1 baseline comparisons, and analysis of BC, GRPO, and DPO failure modes.
- **Mark Music:** Led the full pre-training behavior-cloning stage: the PhysicalAI-AV data pipeline, Gemini 2.5 Flash navigation-label generation with classifier-free guidance dropout, wiring $\pi_{0.5}$'s flow-matching head into the unicycle driving representation, and writing the LoRA adapters and action-expert fine-tuning on 8×H100 that produced $\pi$-Drive-BC.

## 9 AI Tools Disclosure

We used AI tools as development aids. ChatGPT was used for writing assistance, report organization, LaTeX formatting, debugging support, and technical clarification. Claude Code, Codex, and Cursor were used for engineering support across data pipelines, figure-generation scripts, experiment scripts, logging, checkpoint handling, refactoring, and debugging. Mark Music and Felipe Barbosa both used AI coding assistants during parts of the behavior-cloning and post-training workflows.

The core algorithmic work was still designed, implemented, inspected, and verified by the team. In particular, AI tools were not used as a substitute for the essential implementations of the flow-matching training objective, the PPO-style Flow-GRPO update, group-relative advantages, KL anchoring, the DPO objective, or the imitation-anchored DPO stabilization. Mark owned the behavior-cloning retraining pipeline, Felipe owned the post-training algorithms, and Alex Kim owned the evaluation methodology. AI tools accelerated implementation, writing, debugging, and figure creation, but the main experimental design, analysis, and conclusions were developed and verified by the team.

## References


Kevin Black et al. 2025a. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054* (2025). `https://arxiv.org/abs/2504.16054`

Kevin Black, Noah Brown, Danny Driess, et al. 2024. $\pi_0$: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164* (2024).

Kevin Black, Manuel Y. Galliker, and Sergey Levine. 2025b. Real-Time Execution of Action Chunking Flow Policies. *arXiv preprint arXiv:2506.07339* (2025).

Danny Driess, Jost Tobias Springenberg, Brian Ichter, et al. 2025. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. *arXiv preprint arXiv:2505.23705* (2025).

Peter Karkus, Maximilian Igl, Yuxiao Chen, Kashyap Chitta, Jef Packer, Bertrand Douillard, Ran Tian, Alexander Naumann, Guillermo Garcia-Cobo, Shuhan Tan, Alperen Degirmenci, Alexander Popov, Nikolai Smolyanskiy, Urs Muller, Boris Ivanovic, and Marco Pavone. 2025. Beyond Behavior Cloning in Autonomous Driving: A Survey of Closed-Loop Training Techniques. *arXiv preprint* (2025).

Jie Liu et al. 2025. Flow-GRPO: Training Flow Matching Models via Online Reinforcement Learning. *arXiv preprint arXiv:2505.05470* (2025). `https://arxiv.org/abs/2505.05470`

NVIDIA. 2025. NVIDIA PhysicalAI-Autonomous-Vehicles Dataset. Hugging Face dataset. `https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles`

NVIDIA. 2026a. Cosmos 3: Omnimodal World Models for Physical AI. Technical report. `https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf`

NVIDIA. 2026b. NVIDIA TensorRT. `https://developer.nvidia.com/tensorrt`. Accessed 2026-06-07.

Rafael Rafailov et al. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, Vol. 36. `https://arxiv.org/abs/2305.18290`

Bram Wallace et al. 2023. Diffusion Model Alignment Using Direct Preference Optimization. *arXiv preprint arXiv:2311.12908* (2023). https://arxiv.org/abs/2311.12908

Yan Wang et al. 2025. Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail. *arXiv preprint arXiv:2511.00088* (2025). https://arxiv.org/abs/2511.00088