PAPODRIVE: PATTERN-AWARE POLICY OPTIMIZATION FOR SAFE AND EFFICIENT DRIVING VISION-LANGUAGE-ACTION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL)-based post-training has become a mainstream paradigm for improving the safety of vision-language-action (VLA) models in autonomous driving, yet the safety gains often come at the cost of overly conservative behaviors and significant efficiency degradation. To address this safety-efficiency trade-off, we propose *PapoDrive*, a **P**attern-**A**ware **P**olicy **O**ptimization framework for the RL-based post-training. Specifically, we model the VLA planning as a two-stage decision, where the large language model (LLM) first selects a macro reasoning pattern, and the planner then decodes a trajectory conditioned on it. Under this view, the conventional KL-regularized RL essentially acts as a reweighting mechanism over the reference reasoning patterns, with the resulting efficiency drift governed by the covariance between the pattern-level efficiency and safety rewards, which is significantly negative in many driving scenarios and thus becomes the source of this trade-off. Guided with this insight, we develop a pattern-level step-size modulation (PSM) mechanism, allowing efficient patterns to move further from the reference policy while keeping inefficient ones close to it. We further design a lightweight pattern extraction scheme, such that PSM can be integrated into group relative policy optimization (GRPO) with a negligible overhead. PapoDrive improves the *Success Rate* by 5.72 and *Efficiency* by 6.15 over the ORION baseline on Bench2Drive benchmark, and achieves the best safety metrics while limiting efficiency drop to 2.2%, compared with 10.2% for GRPO on high-risk Navsim scenarios.

1 INTRODUCTION

Vision-language-action (VLA) models have achieved a notable success in the field of autonomous driving. Compared with the traditional modular pipelines or end-to-end systems without reasoning (Hu et al., 2023; Jiang et al., 2023), VLA models accumulate the rich world knowledge, thus offering better interpretability and generalization in decision-making (Fu et al., 2025; Xiong et al., 2026). To further improve the safety in complex scenarios, reinforcement learning (RL)-based post-training has become a common practice. Taking the collision feedback as a natural verifiable reward, recent studies (Xiong et al., 2026; Zhou et al., 2026) have brought clear safety gains by adopting the group relative policy optimization (GRPO) algorithm (Shao et al., 2024).

These safety gains, however, often come at sacrifice of driving efficiency (Keswani & Bhattacharyya, 2025; Yang et al., 2026; 2025b; Zhang et al., 2024), yielding overly conservative policies after post-training. We empirically observe this safety-efficiency trade-off consistently across VLA backbones and benchmarks (Figs. 1(a) and Table 2). Taking NAVSIM scenarios of Table 2 for example, GRPO raises the primary safety metric *NC* from 80.2 to 90.4, but reduces efficiency by 10.2%.

A common approach to handling such a trade-off is the joint reward optimization (JRO) (Xiong et al., 2026; Zhou et al., 2026), which incorporates the multiple objectives into an overall reward signal to jointly steer the policy towards a balanced direction. However, we find that the JRO does narrow the efficiency drift, but through an additive efficiency bonus that is granted regardless of the safety. When the efficiency bonus is heavily weighted, it can override the safety preference (see Section 4.1). Therefore, the JRO alone is insufficient to resolve this safety-efficiency trade-off.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

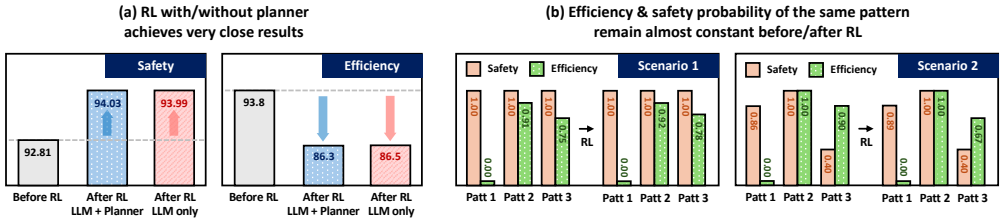


Figure 1: Empirical investigations on the significance of reasoning patterns in driving planning, which are conducted on Bench2Drive benchmark, by employing ORION for inference.

In this paper, we analyze the trade-off from the perspective of reasoning patterns. The planning process of driving VLA can be modeled as a two-stage decision: the LLM first determines a macro logic path, and the planner then decodes a trajectory conditioned on this path (Section 3.1). We call such a macro path a *reasoning pattern*. By deriving a closed-form solution of the optimal policy, we demonstrate that RL post-training is essentially reweighting the probability distribution of model's reasoning patterns (Section 3.2). The resulting efficiency drift, *i.e., the expected efficiency degradation,* is governed by the covariance between the pattern-level safety and efficiency scores (Section 3.3). Guided with this analysis, we propose a pattern-aware policy optimization (PapoDrive) framework that adopts a pattern-level step-size modulation (PSM) mechanism to mitigate the safety-efficiency trade-off (Section 4.2). Unlike the JRO, our PSM does not reward a pattern for being efficient, but only changes the intensity upon which each pattern can be optimized. PSM, which is complementary to JRO, is therefore incorporated into our PapoDrive. In addition, applying PSM in the standard GRPO pipeline requires the patterns to be efficiently identified during training, and we design a lightweight scheme to achieve this (Section 4.3). Our main contributions are as follows:

- **Pattern-level analysis of safety-efficiency trade-off**. We model VLA planning as a two-stage decision of pattern selection followed by trajectory decoding. We derive under this view and show that the efficiency drift is governed by the covariance between pattern-level safety and efficiency.
- **Pattern-aware policy optimization**. Based on theoretical insights, we design a pattern-level step-size modulation (PSM) mechanism to mitigate the trade-off. Implemented with a lightweight pattern extraction scheme, PSM can be integrated into GRPO with a negligible overhead.
- **Empirical evaluations**. We evaluate PapoDrive across different VLA backbones (ORION & ReCogDrive) and benchmarks (Bench2Drive & Navsim). The results demonstrate that PapoDrive consistently mitigates the trade-off, achieving a simultaneous improvement on both the *Success Rate* by +5.72 and *Efficiency* score by +6.15 on the Bench2Drive benchmark.

2 Preliminaries and Related Works

2.1 VLA Model for Autonomous Driving

Driven by the impressive performance of LLMs in contextual comprehension and long-horizon reasoning, recent studies have incorporated Vision-Language-Model (VLM) into autonomous driving. Early attempts were limited mainly to using VLM to assist training of the traditional end-to-end autonomous driving system (VLM-AD (Xu et al., 2024a), VLM-E2E (Liu et al., 2025a)), or providing language explanation for scenarios (DriveGPT4 (Xu et al., 2024b)), rather than direct control.

To bridge this gap between the semantic reasoning space and trajectory action space, various paradigms have been explored. For example, EMMA (Hwang et al., 2024) and ReasonPlan (Liu et al., 2025b) prompted the model to directly output planning trajectories in text, while CarLLaVA (Renz et al., 2024) and AutoVLA (Zhou et al., 2026) discretized the planning results into tokens for auto-regressive output. However, both of these paradigms suffer from insufficient planning precision and physical infeasibility. More recently, the incorporation of generative planners (VAE or Diffusion-based) has led to Vision-Language-Action (VLA) models with a superior closed-loop performance. Prominent examples include ORION (Fu et al., 2025), ReCogDrive (Xiong et al., 2026).

Fig. 2 illustrates a typical architecture of current VLA autonomous driving systems, comprising three modules. First, a visual encoder encodes the raw perceptual signals into embedding vectors, mapping the perception space to the reasoning space. These embeddings are then fed into a Large Language Model (LLM) for the Chain-of-Thought (CoT) reasoning. Finally, the reasoning context is encoded into a latent vector, which guides the subsequent generative planner in trajectory generation.

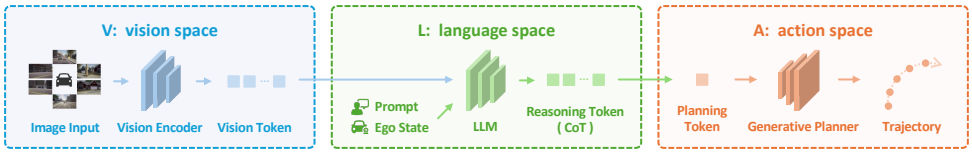


Figure 2: Architecture of the VLA autonomous driving system.

2.2 Reasoning Pattern

To understand the mechanism by which RL post-training improves model's reasoning capability, recent research in LLM community have proposed a theoretical modeling framework from the perspective of "reasoning pattern" (Chen et al., 2026b; Cho et al., 2025; Huang et al., 2025; Jiang et al., 2025). Rather than treating the token-by-token chain-of-thought (CoT) as the unit of analysis, Chen et al. (Chen et al., 2026b) formalized reasoning as a two-stage process: the model first determines a logical path at macro-level, then generates specific answer based on the chosen path. These macro-level paths are named "reasoning patterns", a single problem can be tackled via multiple patterns, and each pattern carries certain probability of leading to successful or failed answer. Their central empirical finding is that RL post-training mainly brought gain by optimizing the distribution of reasoning patterns at the first stage, whereas the per-pattern success rate at the second stage remained relatively stable. This finding identifies pattern, rather than token or individual CoT, as appropriate granularity at which the effect of RL post-training should be analyzed. Building upon this framework, Cho et al. (Cho et al., 2025) further analyzed the trade-off between performance and safety. Inspired by these studies on LLMs, we found that the decision-making process of driving VLA models shares similar two-stage generation nature. Accordingly, we conduct systematic theoretical analysis of the RL post-training for driving VLA models from the perspective of reasoning patterns.

3 Theoretical Analysis of Safety-Efficiency Trade-Off

3.1 Two-Stage Modeling of VLA Planning Process

As discussed in Section 2.1, VLA driving models consist of three components: visual encoder, LLM, and generative planner. Among them, visual encoder is typically kept frozen during post-training, does not participate in the optimization of decision-making. RL post-training therefore acts on the remaining two components: reasoning process of LLM and trajectory decoding process of planner.

In current practice, the planner decodes trajectories within a conditional generation process. It takes the output of LLM's reasoning as condition, and generates concrete trajectory via diffusion or flow-matching (Xiong et al., 2026; Liao et al., 2025). Therefore, once the condition is determined, the output distribution of planner is anchored within a relatively narrow neighborhood, leaving limited space for trajectory variation. Such architecture naturally aligns with the two-stage framework mentioned in Section 2.2, where LLM first determines a macro driving decision and concrete trajectory is then generated along this decision. We designed a set of experiments to verify this intuition:

As illustrated in Fig. 1(a), we compare two RL settings: *i)* jointly optimizing LLM and planner, *ii)* optimizing only the LLM with planner frozen. The performance changes of these two settings are very close, indicating the dominant role of LLM in planning. We further examined where this gain arises in Fig. 1(b), and found that safety / efficiency scores of the same pattern, *i.e.* $s_x(p) = \mathbb{E}[r_{\text{safety}}(\boldsymbol{x}, \boldsymbol{\tau}) \mid p]$ and $g_x(p) = \mathbb{E}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau}) \mid p]$, remain relatively stable before & after RL, indicating that gain of RL indeed comes mainly from optimizing the pattern selection of LLM.

Based on these empirical findings, we formally model the planning process of VLA model as two-stage decision: *i)* LLM performs chain-of-thought (CoT) reasoning to provide a specific reasoning pattern; *ii)* Planner generates trajectory conditioned on the pattern, which can be expressed as:

$$\mathcal{P} = \pi_{\text{LLM}}(\cdot|\boldsymbol{x}),\ \boldsymbol{\tau} \sim \pi_{\text{Planner}}(\cdot|\boldsymbol{x}, p \sim \mathcal{P}), \tag{1}$$

where we denote inputs of VLA by $\boldsymbol{x}$, including multi-view camera images, system prompt, and ego-state (*i.e.,* comprehensive descriptions of the current driving scenario). $\mathcal{P}$ represents the distribution of reasoning pattern p conditioned on $\boldsymbol{x}$, which is instantiated by π_{LLM}. The trajectory $\boldsymbol{\tau}$, conditioned on $p \sim \mathcal{P}$, is decoded by the planner and typically consists of a sequence of future waypoints.

3.2 MECHANISM ANALYSIS OF RL POST-TRAINING

Given this two-stage modeling approach, we conducted further analysis into the nature of RL post-training. To enhance the safety performance of VLA models, the typical RL objective is given by:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, p\sim\pi_\theta(\cdot|\boldsymbol{x}), \boldsymbol{\tau}\sim\pi_{\text{planner}}(\cdot|\boldsymbol{x},p)} \left[r_{\text{safety}}(\boldsymbol{x},\boldsymbol{\tau}) - \beta \cdot \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \tag{2}$$

where $\mathcal{D}$ represents the input dataset, π_θ denotes the policy instantiated by LLM, and π_{ref} is the pre-trained reference policy. $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ and β represents the KL divergence and its coefficient.

We primarily focus on the LLM fine-tuning in this paper, due to its dominant influence on decision-making, whereas the planner keeps frozen. To ensure a fair algorithmic comparison and fundamental exploration, we employed the most concise and straightforward RL objective: binary 0/1 reward, synchronous policy update and without complex treatments like importance sampling or clipping.

RL post-training can be formulated as an optimization problem to maximize the objective in Eq. (2), which yields a theoretical closed-form solution as Proposition 1. See Appendix E for proof.

Proposition 1 (Optimal Policy of RL Post-Training) . *The optimal RL policy π_θ^* can be interpreted as a reweighting of the reference pattern distribution according to the safety scores:*

$$\pi_\theta^*(p \mid \boldsymbol{x}) = \frac{\exp\left(\frac{1}{\beta} s_x(p)\right) \pi_{ref}(p \mid \boldsymbol{x})}{Z(\boldsymbol{x})},\ Z(\boldsymbol{x}) = \sum_{p\in\mathcal{P}} \exp\left(\frac{1}{\beta} s_x(p)\right) \pi_{ref}(p \mid \boldsymbol{x}), \tag{3}$$

where $Z(\boldsymbol{x})$ is the partition function used to normalize the policy into a valid probability distribution.

Definition 1 (Weighting Factor) . *For conciseness in expression, we define the weighting factor $w_x(p) = \exp(\frac{1}{\beta} s_x(p))$, which is coupled with the safety reward probability of reasoning pattern p.*

Proposition 1 reveals that the physical essence of RL post-training is *reweighting the pattern distribution of the reference policy*. The intensity of reweighting is determined by the weighting factor $w_x(p)$. During the safety-oriented post-training, $w_x(p)$ increases with the safety reward probability $s_x(p)$, indicating that patterns yielding greater safety rewards are more likely to be chosen.

3.3 ORIGIN OF SAFETY-EFFICIENCY TRADE-OFF

Given the scenario input $\boldsymbol{x}$, the expectation of efficiency reward probability with the LLM π_θ is formulated as: $\mathbb{E}_{\pi_\theta}[r_{\text{efficiency}}(\boldsymbol{x},\boldsymbol{\tau})] = \sum_{p\in\mathcal{P}} \pi_\theta(p \mid \boldsymbol{x}) \cdot g_x(p)$. We can define the efficiency drift brought by RL in Definition 2, then provide its expression as Proposition 2. Proof in Appendix F.

Definition 2 (Efficiency Drift) . *We define the efficiency drift as the difference in the expected efficiency rewards between π_θ^* and π_{ref}, i.e., $\mathbb{E}_{\pi_\theta^*}[r_{efficiency}(\boldsymbol{x},\boldsymbol{\tau})] - \mathbb{E}_{\pi_{ref}}[r_{efficiency}(\boldsymbol{x},\boldsymbol{\tau})]$.*

Proposition 2 (Expression of Efficiency Drift) . *The efficiency drift is governed by the covariance between efficiency score $g_x(\mathcal{P})$ and weighting factor $w_x(\mathcal{P})$ under reference distribution.*

$$\mathbb{E}_{\pi_\theta^*}[r_{efficiency}(\boldsymbol{x},\boldsymbol{\tau})] - \mathbb{E}_{\pi_{ref}}[r_{efficiency}(\boldsymbol{x},\boldsymbol{\tau})] = \frac{Cov_{\pi_{ref}}(g_x(\mathcal{P}), w_x(\mathcal{P}))}{\mathbb{E}_{\pi_{ref}}[w_x(\mathcal{P})]}. \tag{4}$$

As the covariance $\text{Cov}_{\pi_{\text{ref}}}(g_x(\mathcal{P}), w_x(\mathcal{P}))$ measures the correlation between variables, Proposition 2 suggests that *the direction of efficiency drift is contingent upon how the safety / efficiency rewards are correlated*. We cannot compute the exact values of $\text{Cov}_{\pi_{\text{ref}}}(g_x(\mathcal{P}), w_x(\mathcal{P}))$ for all driving scenarios $\boldsymbol{x}$, so this efficiency drift is not suitable for determining an exact bound. However, it can serve as a theoretical foundation for understanding and managing the trade-off in RL post-training.

We conducted statistical sampling analysis of driving scenarios from Navsim (Dauner et al., 2024) and Bench2Drive (Jia et al., 2024) training sets (Fig. 3(a)), and found that $\text{Cov}_{\pi_{\text{ref}}}(g_x(\mathcal{P}), w_x(\mathcal{P}))$ is significantly negative in approximately 30% of scenarios in Navsim, this proportion rises to 47.7% in Bench2Drive. In such a substantial fraction of scenarios, efficiency / safety rewards drive the optimization toward conflicting directions, which constitutes the origin of the trade-off observed in RL post-training. Representative examples are illustrated in Fig. 3(c). We further observed that these negatively correlated scenarios tend to involve more complex traffic interactions (Fig. 3(b)), which highlights the necessity to address trade-off during the post-training of reasoning capability.

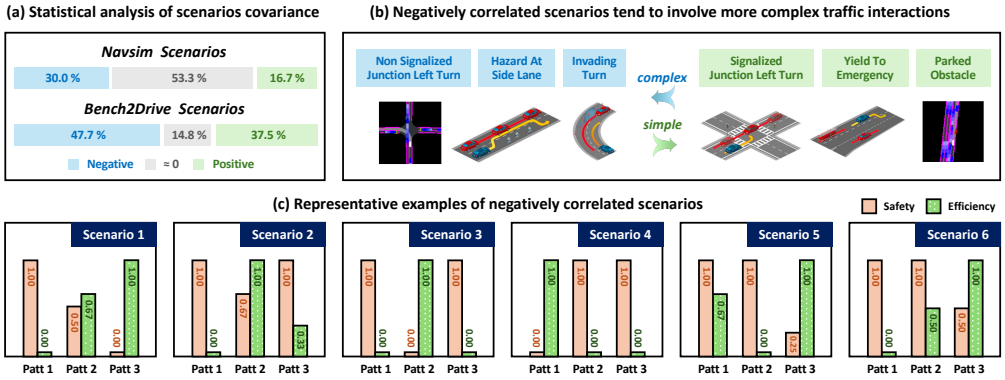


Figure 3: Statistical analysis of driving scenarios in mainstream benchmarks.

4 PROPOSED PAPODRIVE

4.1 REVISITING JOINT REWARD OPTIMIZATION (JRO)

Guided by the theoretical framework established in Section 3, the commonly adopted joint reward optimization approach can be examined from a more fundamental perspective. Given its objective:

$$\mathcal{J}_{\mathrm{RL}}^{\mathrm{JRO}}(\theta)=\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x}),\boldsymbol{\tau}\sim\pi_{\mathrm{planner}}(\cdot|\boldsymbol{x},p)}\left[\, r_{\mathrm{safety}}(\boldsymbol{x},\boldsymbol{\tau})+\lambda\cdot r_{\mathrm{efficiency}}(\boldsymbol{x},\boldsymbol{\tau})-\beta\cdot\mathbb{D}_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}})\right], \quad (5)$$

a similar derivation can be followed to obtain the corresponding solution of the optimal policy:

$$\pi^*_{\theta-\mathrm{JRO}}(p\mid\boldsymbol{x})=\frac{\exp\left(\frac{1}{\beta}(s_x(p)+\lambda\cdot g_x(p))\right)\pi_{\mathrm{ref}}(p\mid\boldsymbol{x})}{Z(\boldsymbol{x})}, \quad (6)$$

with $Z(\boldsymbol{x})$ omitted for brevity. Compared with the safety-oriented optimal policy in Eq. (3), the essential distinction lies in the introduction of an efficiency score term into the weighting factor:

$$w_x(p)=\exp\left(\frac{s_x(p)}{\beta}\right)\Rightarrow w_x^{\mathrm{JRO}}(p)=\exp\left(\frac{s_x(p)+\lambda\cdot g_x(p)}{\beta}\right), \quad (7)$$

where λ is the coefficient of efficiency reward in objective Eq. (4). As established in Section 3.3, the negative correlation between $g_x(\mathcal{P})$ and $w_x(\mathcal{P})$ leads to a significantly negative covariance in many scenarios, which is the origin of trade-off. Substituting the new $w_x^{\mathrm{JRO}}(p)$ into Eq. (4), it can be found that the introduced efficiency term weakens this inherent negative correlation, thereby tightening the magnitude of the efficiency drift. Up to this point, our framework provides a pattern-level theoretical justification for why joint reward optimization is able to mitigate the trade-off.

However, there are also two defects revealed: *i)* The introduced efficiency term $\lambda\cdot g_x(p)$ acts as an "additive bonus", granted irrespective of whether a pattern is safe. A fast pattern that always collides ($s_x(p)=0$) will also be encouraged. *ii)* When λ is large, the safety rank among patterns can be reversed by efficiency rank, increasing the probability of unsafe patterns during optimization, which may even induce a reverse trade-off. These two defects combine into a dilemma: scenarios with the strongest safety-efficiency conflict require the largest λ, yet a large λ is precisely what makes pattern-rank reversal more likely. The benefit and side effect are thus entangled inherently.

4.2 PATTERN-LEVEL STEP-SIZE MODULATION (PSM)

Motivated by the analysis above, we proposed another approach to tightening the efficiency drift. Instead of adding efficiency to the reward, we let the strength of the KL constraint depend on the efficiency of each pattern, through an adjustment function $m_x(p)$:

$$\mathcal{J}_{\mathrm{RL}}^{\mathrm{PSM}}(\theta)=\mathbb{E}_{x,p,\tau}\left[r_{\mathrm{safety}}(x,\tau)-\beta\, m_x(p)\log\frac{\pi_\theta(p|x)}{\pi_{\mathrm{ref}}(p|x)}\right] \quad (8)$$

$$m_x(p)=\frac{1+\alpha\cdot(1-g_x(p))}{1+\gamma\cdot g_x(p)}, \quad (9)$$

where α and γ are positive parameters. The adjustment by $m_x(p)$ carries clear interpretation:

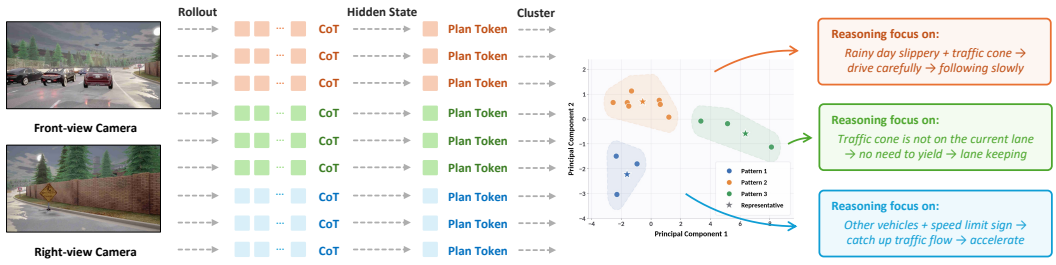


Figure 4: Illustration of the reasoning pattern extraction process used in PapoDrive.

- **For high-efficiency patterns**, $g_x(p) \approx 1$, $m_x(p) \approx 1/(1+\gamma)$. The weighting factor is amplified, encouraging the probability increase of high-efficiency patterns.
- **For low-efficiency patterns**, $g_x(p) \approx 0$, $m_x(p) \approx 1+\alpha$. The weighting factor is diminished, suppressing the probability increase of low-efficiency patterns.

Proposition 3 (Optimal Policy of PSM) *Given x, the objective of PSM has the unique maximizer*

$$\pi^*_{\theta\text{-}PSM}(p \mid x) = \pi_{\text{ref}}(p \mid x)\, \exp\Big(\frac{s_x(p) - \mu^*_x}{\beta\, m_x(p)} - 1\Big), \tag{10}$$

where μ^*_x is the unique root of $\sum_p \pi_{\text{ref}}(p|x) \exp\big(\frac{s_x(p)-\mu}{\beta m_x(p)} - 1\big) = 1$, and it satisfies $\min_p \tilde{s}_x(p) \le \mu^*_x \le \max_p \tilde{s}_x(p)$ with $\tilde{s}_x(p) := s_x(p) - \beta\, m_x(p)$.

Proposition 3 (See proof in Appendix G) makes the role of $m_x(p)$ explicit. Define the centered advantage $A^*_x(p) = s_x(p) - \mu^*_x$. Then $\log\big(\pi^*_{\theta\text{-PSM}}/\pi_{\text{ref}}\big) = A^*_x(p)/(\beta m_x(p)) - 1$, so each pattern moves away from the reference policy by a step whose size is proportional to $1/m_x(p)$. The direction of the step is set by safety, and $m_x(p)$ sets how far the pattern may move. In the Proposition 2 form, the weighting factor becomes $w^{\text{PSM}}_x(p) = \exp\big(A^*_x(p)/(\beta m_x(p))\big)$.

Proposition 4 (Safety-Order Preservation) *A pattern p is up-weighted ($\pi^*_{\theta\text{-}PSM}(p|x) > \pi_{\text{ref}}(p|x)$) if and only if $\tilde{s}_x(p) > \mu^*_x$. Consequently, if $s_x(p) - s_x(q) > \beta\,(m_{\max} - m_{\min}) = \beta(\alpha + \frac{\gamma}{1+\gamma})$, it cannot happen that q is up-weighted while p is not.*

With $\beta = 0.1, \alpha = 1.5, \gamma = 0.5$, the threshold is 0.183. The efficiency-induced shift is therefore bounded by a constant set by the KL scale. By contrast, the additive bonus of JRO shifts the score by $\lambda\,(g_x(p) - g_x(q))$, which grows with λ.

We observed in experiment that safety-oriented GRPO causes the policy rapidly to concentrate on conservative distribution in Fig. 5(b), whereas PSM can suppress this over-concentration during the early stages of training, preserving the diversity of fast patterns. Therefore, PSM is suitable as an orthogonal approach to be used together with joint reward optimization: the former regulates the optimization dynamics, while the latter shapes the optimization target.

4.3 Pattern Extraction

The last obstacle to integrate PSM into the standard GRPO pipeline is: how to efficiently extract and represent patterns. Current extraction schemes include rule-based parsing, keyword matching, or an auxiliary LLM to label each CoT (Chen et al., 2026b). However, these schemes suffer from a high customization requirement and high processing latency, making them unsuitable for large-scale deployment in the training of VLA models. Consequently, we propose a pattern extraction scheme specifically tailored for the autonomous driving VLA models with low computational overhead.

As illustrated in Fig. 2, the LLM outputs a planning token after completing the reasoning, which is typically the last hidden state vector obtained from the final forward propagation. This vector encodes the previous scenario and reasoning content and is provided to the planner as condition for trajectory decoding. Therefore, we found that the planning token is a suitable proxy that can be directly clustered in the semantic space to obtain different patterns.

Specifically, the extraction process is conducted via two steps: *i)* we adopt principal component analysis (PCA) to suppress noise and obtain the principal components $\mathbf{Z}$ of *planning tokens*; and *ii)* a K-Means clustering $\mathcal{P} = K\text{-Means}(\mathbf{Z})$ is utilized to automatically categorize the *planning tokens* into representative reasoning patterns. Fig. 4 presents an example of pattern extraction. As shown, the resulting patterns are indeed semantically meaningful, reasoning the same scenario from different perspectives. The whole process adds negligible overhead, as it operates on vectors already computed during rollout. Further discussion of pattern extraction is provided in Section 5.3. The overall algorithm of our proposed Pattern-Aware Policy Optimization (PapoDrive) is illustrated in Algorithm 1 of Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

i) VLA backbone and benchmarks. To verify the generalization capacity, we validate PapoDrive across different VLA backbones (ORION (Fu et al., 2025) & ReCogDrive (Xiong et al., 2026)) and benchmarks (Bench2Drive (Jia et al., 2024) & Navsim (Dauner et al., 2024)). See Appendix A for details. ***ii) Training settings.*** Post-training is conducted on 8 NVIDIA H800 GPUs. $G = 8$ rollouts for each scenario, with high temperature to ensure diversity. A base value $\beta = 0.1$ is applied for KL penalty. Other settings (like batchsize, *etc*.) vary for different backbone, reported in Appendix B.

5.2 MAIN RESULTS

Performance on Bench2Drive. Performance of PapoDrive on Bench2Drive closed-loop evaluation is reported in Table 1. Taking the latest state-of-the-art (SOTA) scheme ORION as a baseline, PapoDrive enhances the *Success Rate* (SR) by $+5.72$ and the *Efficiency* (Eff.) by $+6.15$, attaining a desirable simultaneous optimization on both safety and efficiency. Through the PapoDrive post-training, we have established a new record for *Success Rate* and *Driving Score* (DS) on the Bench2Drive leaderboard. In terms of efficiency, ReasonPlan (Liu et al., 2025b) and DriveDPO (Shang et al., 2026) exhibit highly aggressive driving policies. However, a direct comparison remains inequitable due to their fundamental differences from ORION in terms of policy backbones and trajectory generation mechanisms (generative planner vs. direct text output). As a post-training algorithm, the most direct evidence of PapoDrive's effectiveness lies in its comparison with the baseline scheme, ORION. Closed-loop evaluation has significant variance, so we report the Error Bar from multiple evaluations to demonstrate that the proposed method can bring stable improvements.

Multi-ability results on Bench2Drive. In terms of multi-ability results, the largest gains appear in *Merge* ($+16.43$) and *GiveWay* ($+20.00$), which involve dense interactions with other agents and coincide with the scenario types where we observe negatively correlated safety-efficiency rewards in Fig. 3(b). Our method falls behind ORION in *Traffic-Sign* (T.Sign). This limitation suggests that focusing solely on collision-based safety metrics may neglect some wider safety aspects in driving, such as compliance with traffic signs. Incorporating a more diverse range of verifiable rewards into the post-training process represents a key direction for our future engineering optimizations.

Performance on Navsim (full). On Navsim benchmark, we applied post-training based on the advanced method ReCogDrive, with results reported in Table 6 (report in Appendix C). PapoDrive improves all three primary safety metrics: *NC*, *DAC* and *TTC*. However, most test scenarios in Navsim are relatively simple, with safety metrics already close to saturation, leaving limited room to show the improvement brought by post-training. We therefore conducted more focused evaluation and analysis on the most challenging high-risk scenarios, see Table 2.

Performance on Navsim (high-risk). Vanilla GRPO (ID-2) exhibits a severe trade-off, which improves *NC* from 80.2 to 90.4 and *TTC* from 70.8 to 81.9, but reduces efficiency from 137 to 123 (-10.2%). This confirms the analysis in Section 3.3: safety and efficiency are frequently negatively correlated in complex high-risk scenarios. Joint reward optimization (ID-3) maintains efficiency, but the safety gain almost vanishes, even drops below the base model. This is consistent with the defects identified in Section 4.1, the additive efficiency bonus interferes with optimization of safety. In comparison, PapoDrive (ID-5) mitigates the trade-off with pattern-level step-size modulation, achieving higher safety metrics than vanilla GRPO, while the efficiency drop is limited to 3 (-2.2%). Fig. 5(a) compares the safety-efficiency Pareto fronts of GRPO and PapoDrive across all saved checkpoints

Table 1: **Planning results on Bench2Drive closed-loop benchmark.** † represents models trained on PDM-Lite, which provides higher-quality expert demonstrations than official Bench2Drive (Think2Drive). Consequently, these models don't constitute fair comparison and are included for reference only. Bold indicates the best, and underline denotes the second best.

Method	Closed-loop Metric ↑			Multi-Ability Metric (%) ↑					
	Eff.	SR	DS	Merge	Overtake	E.Brake	GiveWay	T.Sign	Mean
• *Trained on PDM-Lite expert*									
AutoVLA† (Zhou et al., 2026)	146.93	57.73	78.84	–	–	–	–	–	–
SimLingo† (Renz et al., 2025)	244.18	66.82	85.94	54.01	57.04	88.33	53.33	82.45	67.03
R2SE† (Liu et al., 2026)	243.89	69.54	86.28	53.33	61.25	90.00	50.00	84.21	67.76
• *Trained on Think2Drive expert*									
TCP-traj (Wu et al., 2022)	76.54	30.00	59.90	8.89	24.29	51.67	40.00	46.28	34.22
AD-MLP (Zhai et al., 2023)	48.45	00.00	48.45	00.00	00.00	00.00	00.00	4.35	00.87
UniAD-Base (Hu et al., 2023)	129.21	16.36	45.81	14.10	17.78	21.67	10.00	14.21	15.55
VAD (Jiang et al., 2023)	157.94	15.00	42.35	8.11	24.44	18.64	20.00	19.15	18.07
ThinkTwice (Jia et al., 2023b)	69.33	31.23	62.44	27.38	18.42	35.82	**50.00**	54.23	37.17
DriveAdapter (Jia et al., 2023a)	70.22	33.08	64.22	28.82	26.38	48.76	**50.00**	56.43	42.08
GenAD (Zheng et al., 2024)	–	15.90	44.81	–	–	–	–	–	–
DriveDPO (Shang et al., 2026)	166.80	30.62	62.02	–	–	–	–	–	–
DriveTrans. (Jia et al., 2025)	100.64	35.01	63.46	17.57	35.00	48.36	40.00	52.10	38.60
ReasonPlan (Liu et al., 2025b)	**180.64**	34.55	64.01	37.50	26.67	33.30	40.00	45.79	36.66
ReAL-AD (Lu et al., 2025)	–	11.36	41.17	–	–	–	–	–	–
RAP (Feng et al., 2026)	165.47	37.27	66.42	–	–	–	–	–	–
ReCogDrive (Xiong et al., 2026)	138.18	45.45	71.36	29.73	20.00	69.09	20.00	**71.34**	42.03
ORION (Fu et al., 2025) *(base)*	151.48	54.62	77.74	25.00	71.11	78.33	30.00	69.15	54.72
PapoDrive (best-of-n)	160.18	**61.03**	**85.16**	**41.43**	**76.74**	**83.33**	**50.00**	54.74	**59.42**
Δ *vs. ORION*	+8.70	+6.41	+7.42	+16.43	+5.63	+5.00	+20.00	-14.41	+4.70
PapoDrive (mean)	157.63	60.34	84.92	37.91	71.83	80.55	43.33	53.72	57.47
Error Bar (Std)	±2.41	±0.76	±0.38	±4.26	±5.04	±2.52	±5.77	±1.00	±1.81

Table 2: **Planning results on Navsim high-risk scenarios.** Consistently report the best-of-n metric over multiple checkpoints for comparison across different post-training settings.

ID	Recog-2B *(base)*	Vanilla GRPO	Joint Reward	Regular Adap.KL	**Pattern Step-size**	Eff.	ΔEff.	NC	ΔNC	DAC	TTC	Comf.	EP	**PDMS**
1	✓	–	–	–	–	137	*base*	80.2	*base*	75.5	70.8	100	50.7	50.7
2	✓	✓	–	–	–	123	-14	90.4	+10	74.3	81.9	100	52.8	58.0
3	✓	✓	✓	–	–	138	+1	80.3	+0.1	74.6	69.5	100	49.6	49.8
4	✓	✓	✓	✓	–	**139**	+2	77.6	-2.6	74.9	69.2	100	47.6	47.9
5	✓	✓	✓	–	✓	134	-3	**91.8**	+11	**78.1**	**83.5**	100	**57.3**	**62.2**

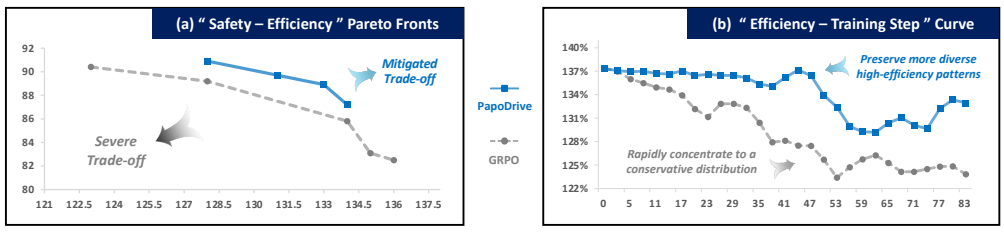


Figure 5: Pareto fronts and Efficiency curves during post-training process on Navsim.

during an identical post-training process. The front of PapoDrive consistently lies above that of GRPO, showing the mitigated trade-off. To highlight the advantage of modulation on the pattern level, we further compare with conventional adaptive KL method (ID-4), which adjusts the KL

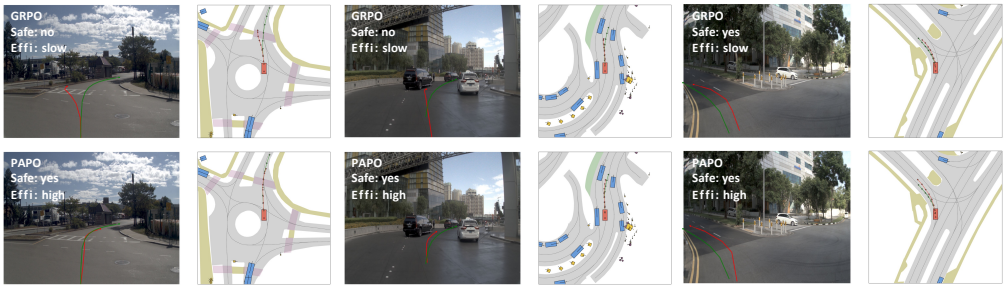


Figure 6: **Qualitative results.** Green: human expert trajectories. Red: VLA model trajectories.

Table 3: Ablation on PSM.

α	γ	Collision ↓	Efficiency ↑
1.5	0.5	5.945	0.8742
0.5	0.5	6.191	0.8601
0.5	1.5	6.101	0.8625

Table 4: Ablation on cluster.

Number	Collision ↓	Efficiency ↑
2	6.058	0.8721
3	5.945	0.8720
4	6.093	0.8707

Table 5: Pattern estimation.

Metric	$G = 8$	$G = 16$
Purity ↑	0.94	0.93
MAE ↓	0.27	0.20
Direction ↑	97.6%	97.0%

penalty based only on the global divergence between the trained and reference policies, unable to resolve the inherent problem of JRO.

Qualitative results. Fig. 6 presents qualitative comparison between GRPO and PapoDrive (PAPO). In the first two scenarios, PapoDrive simultaneously improved the safety and efficiency of planning. In the third scenario, PapoDrive achieved higher efficiency under the same safety premise.

5.3 Ablation Study

Impact of adjustment function. The adjustment function $m_x(p)$ in PSM involves two hyperparameters, α and γ, which control the suppression of low-efficiency patterns and the encouragement of high-efficiency patterns, respectively. As Table 3 shown, the configuration with stronger suppression ($\alpha = 1.5$, $\gamma = 0.5$) achieves the best safety and efficiency, adopted in our final experiments. This is consistent with analysis in Section 4.2: the exponential reweighting rapidly concentrates the policy onto conservative distribution, calling for stronger constraint on low-efficiency patterns.

Impact of clustering algorithm and number. As shown in Table 4, the post-training performance is not sensitive to the number of clusters in K-Means. We further implemented an alternative clustering scheme based on Gaussian Mixture Model (GMM) with Bayesian Information Criterion (BIC), which adaptively determines the number of clusters according to the distribution of sampled planning tokens. Experiments show that GMM+BIC also yields three clusters in most scenarios, and obtains highly consistent partitions with K-Means ($K = 3$). More details in Appendix D.2.

Impact of efficiency score estimation. In PSM, the efficiency score $g_x(p)$ of each pattern is estimated from only a few rollouts, which may introduce estimation noise. To quantify its impact, we sampled $G = 24$ rollouts for each of 1,163 high-risk Navsim scenarios as reference, then bootstrap subsets of $G = 8$ and $G = 16$ rollouts (100 times per scenario), re-running the complete pipeline of clustering, score estimation and modulation. As shown in Table 5, the clustering consistency is stable (purity of 0.94). While the value of $g_x(p)$ is indeed noisy (MAE of 0.27 at $G = 8$), the gap of $g_x(p)$ between patterns (median 0.57) is well above the noise level, so the modulation direction is preserved in 97.6% of trials at $G = 8$, indicating limited impact of estimation noise on PSM.

6 Conclusion

We have focused on the safety-efficiency trade-off during the RL post-training of VLA driving system. By modeling VLA planning as a two-stage process of reasoning pattern selection and trajectory decoding, we showed that efficiency drift is governed by the covariance between safety and efficiency. Guided by these insights, we proposed PapoDrive, which modulates the pattern-level optimization step-size to mitigate trade-off. Empirical evaluations on Bench2Drive and Navsim benchmark validated both our theoretical analysis and algorithmic design.

AI USE STATEMENT

Large Language Models (LLMs) were used to assist with the writing and polishing of the manuscript, including sentence rephrasing, grammar checking, and improving clarity and readability. The LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, implementations, experiments, analyses, and scientific conclusions were developed and conducted by the authors. The authors reviewed all AI-assisted text and take full responsibility for the final content of the manuscript.

REFERENCES

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil is in narrow policy: Unleashing exploration in driving vla models. *arXiv preprint arXiv:2603.06049*, 2026a.

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in LLMs: A theoretical analysis of RL training dynamics through pattern selection. In *International Conference on Learning Representations*, volume 2026, pp. 15084–15113, 2026b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023.

Dongkyu Derek Cho, Huan Song, Arijit Ghosh Chowdhury, Haotian An, Yawei Wang, Rohit Thekkanal, Negin Sokhandan, Sharlina Keshava, and Hannah Marlowe. Breaking the safety-capability tradeoff: Reinforcement learning with verifiable rewards maintains safety guardrails in llms. *arXiv preprint arXiv:2511.21050*, 2025.

OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. `https://github.com/OpenDriveLab/OpenScene`, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 2026.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24823–24834, 2025.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, Zachary Yahn, Yichang Xu, and Ling Liu. Safety tax: Safety alignment makes your large reasoning models less reasonable. *arXiv preprint arXiv:2503.00555*, 2025.

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.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7953–7963, 2023a.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21983–21994, 2023b.

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.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2025, pp. 67227–67243, 2025.

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

Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 23303–23320, 2025.

Mahesh Keswani and Raunak Bhattacharyya. Learning safe autonomous driving policies using predictive safety representations. *arXiv preprint arXiv:2512.17586*, 2025.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, LU HOU, Lue Fan, and Zhaoxiang Zhang. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026.

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

Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Pei Liu, Haipeng Liu, Haichao Liu, Xin Liu, Jinxin Ni, and Jun Ma. Vlm-e2e: Enhancing end-to-end autonomous driving with multimodal driver attention fusion. *arXiv preprint arXiv:2502.18042*, 2025a.

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*, 2025b.

Yuhang Lu, Jiadong Tu, Yuexin Ma, and Xinge Zhu. Real-ad: Towards human-like reasoning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27783–27793, 2025.

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.

Katrin Renz, Long Chen, Ana-Maria Marcu, Jan Hünermann, Benoit Hanotte, Alice Karnsund, Jamie Shotton, Elahe Arani, and Oleg Sinavski. Carllava: Vision language models for camera-only closed-loop driving. *arXiv preprint arXiv:2406.10165*, 2024.

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

Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and ZHAO-XIANG ZHANG. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 38:81565–81585, 2026.

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.

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

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

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*, 2024a.

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*, 9(10):8186–8193, 2024b.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025a.

Rui Yang, Lei Zheng, Shuzhi Sam Ge, and Jun Ma. Safe and nonconservative contingency planning for autonomous vehicles via online learning-based reachable set barriers. *IEEE Transactions on Control Systems Technology*, 2026.

Shuo Yang, Shizhen Li, Yanjun Huang, and Hong Chen. A safe self-evolution algorithm for autonomous driving based on data-driven risk quantification model. *Accident Analysis & Prevention*, 214:107941, 2025b.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

Zheng Zhang, Qi Liu, Yanjie Li, Ke Lin, and Linyu Li. Safe reinforcement learning in autonomous driving with epistemic uncertainty estimation. *IEEE Transactions on Intelligent Transportation Systems*, 25(10):13653–13666, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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

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

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, and Jie Shao. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Appendix

We organize the appendix material as follows. We report the detail information of benchmarks in Appendix A. In Appendix B, we detail our training details. In Appendix C, we provide the planning results on Navsim benchmark. Appendix D gives the detail of ablation experiments. Finally, Appendix E, Appendix F and Appendix G presents the derivation process of our propositions.

A BENCHMARKS

A.1 EAVLUATION METRICS

Bench2Drive Metrics. For closed-loop evaluation, we evaluate the proposed method on the official set of 220 short routes designed by Bench2drive (Jia et al., 2024), spanning 44 interactive scenarios with 5 routes per scenario. Bench2drive includes 3 metrics for closed-loop evaluation:

Success Rate (SR). A route is regarded as successful if the ego vehicle reaches its destination within the allotted time without any traffic infraction. The success rate is the proportion of successful routes among all routes, which is thus a primary indicator of safety:

$$\mathrm{SR} = \frac{n_{\text{success}}}{n_{\text{total}}}, \tag{11}$$

where n_{success} and n_{total} denote the number of successful routes and the total number of routes, respectively.

Driving Score (DS). The driving score jointly considers route completion and infraction penalties. For each route, the completed percentage of the route distance is multiplied by the penalty factors of all infractions that occur, and the results are averaged over all routes:

$$\mathrm{DS} = \frac{1}{n_{\text{total}}} \sum_{i=1}^{n_{\text{total}}} \mathrm{RC}_i \cdot \prod_{j=1}^{n_{i,\text{penalty}}} p_{i,j}, \tag{12}$$

where RC_i is the route completion percentage of the i-th route, and $p_{i,j} \in (0, 1)$ is the penalty factor of the j-th infraction on the i-th route, whose value depends on the infraction severity (e.g., collisions with pedestrians, vehicles or static objects, and running red lights or stop signs).

Efficiency. Efficiency measures whether the ego vehicle keeps up with the surrounding traffic. At each check, the speed percentage is computed as the ratio between the ego speed and the average speed of nearby vehicles:

$$\text{Speed Percentage} = \frac{\text{Ego Vehicle's Speed}}{\text{Average Speed of Nearby Vehicles}}. \tag{13}$$

A value above 100% indicates that the ego vehicle drives faster than nearby traffic. Bench2Drive performs the check every 5% of the route length, i.e., 20 checkpoints per route, and excludes it from the driving score. The final driving efficiency is the average speed percentage over all checks:

$$\text{Driving Efficiency} = \frac{\sum_i \text{Speed Percentage}_i}{\text{Speed Check Times}}. \tag{14}$$

Further, Bench2drive summarize 5 advanced skills for urban driving: Merging, Overtaking, Give Way, Traffic Sign and Emergency Brake and report the score of each skill as a Multi-Ability metrics table, reflecting a more detailed system performance.

Navsim Metrics. The Navsim (Dauner et al., 2024) benchmark provides a non-reactive simulation environment and employs the Predictive Driver Model Score (PDMS) as its planning metric:

$$\text{PDMS} = NC \times DAC \times \left(\frac{5 \times EP + 5 \times TTC + 2 \times C}{12} \right), \tag{15}$$

where PDMS integrates five sub-metrics: No At-Fault Collision (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (Comf.), and Ego Progress (EP) to produce a comprehensive planning score.

The official metrics of Navsim (Dauner et al., 2024) mainly focus on safety, rule compliance and comfort, and do not include a metric dedicated to driving efficiency. Therefore, following the efficiency metric of Bench2Drive (Jia et al., 2024), we define an efficiency metric (Eff.) for Navsim which is the ratio between the ego speed of the planned trajectory and the speed of nearby vehicles:

$$\text{Speed Percentage} = \frac{\text{Ego Vehicle's Speed}}{\text{Average Speed of Nearby Vehicles}}. \tag{16}$$

A value above 100% indicates that the ego vehicle drives faster than nearby traffic. The final efficiency is the average speed percentage over all evaluated scenarios:

$$\text{Eff.} = \frac{\sum_i \text{Speed Percentage}_i}{\text{Number of Scenarios}}. \tag{17}$$

A.2 Construction of Navsim High-Risk Subset

Navsim (Dauner et al., 2024) provides a planning-oriented driving dataset derived from OpenScene (Contributors, 2023). Each sample contains eight cameras with a resolution of 1920×1080, as well as a LiDAR point cloud fused from five sensors over the current and three preceding frames. The dataset is divided into the `navtrain` split for training and the `navtest` split for evaluation.

As discussed in Section 5.3, most test scenarios in Navsim are relatively simple, and their safety metrics are already close to saturation, which leaves limited room to reveal the effect of post-training. To enable a more focused evaluation, we construct a high-risk subset of Navsim. Specifically, we evaluate the publicly released checkpoint of ReCogDrive (Xiong et al., 2026) on all scenarios of the `navtest` split, and retain the scenarios in which either the No at-fault Collision (NC) or the Drivable Area Compliance (DAC) score is not perfect. Since NC and DAC act as multiplicative penalty terms in PDMS, an imperfect score directly indicates a safety-critical failure of the base model, such as an at-fault collision or a departure from the drivable area. This procedure yields 546 scenarios, which constitute the Navsim (high-risk) subset reported in Table 2. All post-training settings in Table 2 are evaluated on this identical subset.

Following the same procedure, we further evaluate the ReCogDrive checkpoint on the `navtrain` split and obtain 1,163 high-risk scenarios, which are used for the analysis of efficiency score estimation in Section 5.3.

A.3 Training Datasets

For post-training, we use the official training sets provided by Bench2Drive (Jia et al., 2024) and Navsim (Dauner et al., 2024). For the perceptual inputs, we follow the standard configuration of each base model: ORION (Fu et al., 2025) takes images from six surround-view cameras as input, whereas ReCogDrive (Xiong et al., 2026) takes only the front-view camera image as input.

B Implementation Details

B.1 Implementation Details of ORION Post-Traing

Following ORION (Fu et al., 2025), we adopt EVA-02-L (Fang et al., 2024) as the vision encoder. Vicuna v1.5 (Zheng et al., 2023) is employed and fine-tuned using LoRA (Hu et al., 2022), with the rank dimension and alpha set to 16. The model is post-trained on 8 NVIDIA H800 GPUs. We use a per-GPU batch size of 1 and accumulate gradients over 4 steps, resulting in an effective batch size of 32. We set the learning rate to 2×10^{-4}, with warm-up and cosine-annealing scheduler, and sample $G = 8$ outputs per frame, with temperature $= 1.0$ to ensure diversity. A base value $\beta = 0.1$ is applied for KL penalty.

B.2 Implementation Details of ReCogDrive Post-Traing

Following ReCogDrive (Xiong et al., 2026), we choose InternVL3-2B (Zhu et al., 2025), comprising a 300M-parameter InternViT visual encoder (Chen et al., 2024) and a Qwen2.5 (Yang et al., 2025a), as the base model. Lora (Hu et al., 2022) rank dimension and alpha are set to 16. The model is post-trained on 8 NVIDIA H800 GPUs. We use a per-GPU batch size of 24, resulting in an effective batch

Algorithm 1: Proposed PapoDrive

Input: Dataset $\mathcal{D}$, training model π_θ, reference policy π_{ref}, hyperparameters λ, α, γ, β_{base}

for *each input scenario* $\boldsymbol{x}$ *in* $\mathcal{D}$ **do**
  Sample G outputs $\{cot_1, \ldots, cot_G\}$ and $\{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_G\}$ using π_θ;
  Compute efficiency and safety reward $r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})$, $r_{\text{safety}}(\boldsymbol{x}, \boldsymbol{\tau})$ for each output;
  Extract reasoning patterns $\mathcal{P}$ from the planning token of each cot_i;
  for *each pattern* p_i **do**
    Calculate efficiency reward probability $g_x(p) = \mathbb{E}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau}) \mid p]$;
    Compute adjustment factor $m_x(p_i)$ following Eq. (9) and set $\beta_i = \beta_{base} \cdot m_x(p_i)$;
  end
  Calculate JRO and PSM losses with Eq. (5) and Eq. (8), then update parameters.
end

Table 6: **Planning results on Navsim benchmark.** PapoDrive is post-trained on ReCogDrive-Base.

Method	Type	LLM Backbone	NC	DAC	TTC	Comf.	EP	PDMS
UniAD-Base (Hu et al., 2023)	Non-VLA	–	97.8	91.9	92.9	100	78.8	83.4
PARA-Drive (Weng et al., 2024)	Non-VLA	–	97.9	92.4	93.0	99.8	79.3	84.0
TransFuser (Chitta et al., 2023)	Non-VLA	–	97.7	92.8	92.8	100	79.2	84.0
DiffusionDrive (Liao et al., 2025)	Non-VLA	–	98.2	96.2	94.7	100	82.2	88.1
AutoVLA (Zhou et al., 2026)	VLA-based	Qwen2.5-VL-3B	98.4	95.6	98.0	99.9	81.9	89.1
ReCogDrive-Large (Xiong et al., 2026)	VLA-based	InternVL3-8B	98.2	97.8	95.2	100	83.5	89.6
DriveVLA-W0 (Li et al., 2026)	VLA-based	Emu-3-8B	98.7	99.1	97.1	100	87.6	90.2
Curious-VLA (Chen et al., 2026a)	VLA-based	Qwen2.5-VL-3B	98.4	96.9	97.9	98.1	88.5	90.3
AdaThinkDrive (Luo et al., 2025)	VLA-based	InternVL3-8B	98.4	97.8	95.2	100	84.4	90.3
ReCogDrive-Base (Xiong et al., 2026)	VLA-based	InternVL3-2B	98.2	97.8	95.4	100	86.0	90.8
PapoDrive	VLA-based	InternVL3-2B	99.0	98.0	96.7	100	85.5	90.8

size of 192. We set the learning rate to 6×10^{-4}, with warm-up and cosine-annealing scheduler, and sample $G = 8$ outputs per frame, with temperature $= 0.7$ to ensure diversity. A base value $\beta = 0.1$ is applied for KL penalty.

B.3 Verifiable rewards Implentation

We construct the verifiable rewards during training as follows. For the safety reward, we employ a straightforward collision-based Boolean criterion: scenarios with collisions are assigned $r_{\text{safety}}(\boldsymbol{x}, \boldsymbol{\tau}) = 0$, while collision-free scenarios are assigned $r_{\text{safety}}(\boldsymbol{x}, \boldsymbol{\tau}) = 1$. For the efficiency reward, inspired by the concept of group relative comparison Shao et al. (2024), we assign $r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau}) = 0$ to samples whose *Effi.* falls below the mean of G samples, and $r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau}) = 1$ otherwise.

B.4 Pseudo-code of PapoDrive Algorithm

The overall algorithm of our proposed Pattern-Aware Policy Optimization (PapoDrive) is illustrated in Algorithm 1.

C Planning results on Navsim Benchmark.

As discussed in Section 5.3, we evaluated our method on the full `navtest` split, with results reported in Table 6.

D Detail of Ablations

D.1 Detail of Efficiency Score Estimation Analysis

In PSM, the efficiency score $g_x(p)$ of each reasoning pattern is estimated from the rollouts assigned to that pattern within a group. With $G = 8$ rollouts and $K = 3$ patterns, the median pattern contains only 3 rollouts. This section details how we quantify the impact of such small-sample estimation.

D.1.1 Experimental Protocol

Reference statistics. We collect 1,163 high-risk scenarios from the `navtrain` split according to Appendix A.2. For each scenario, we sample $G = 24$ rollouts from the reference policy and run the complete pattern extraction pipeline on them, including PCA, K-Means clustering ($K = 3$), estimation of $s_x(p)$ and $g_x(p)$, and computation of $m_x(p)$. The resulting statistics serve as reference.

Subsampling. To simulate the training condition, we randomly subsample $G = 8$ or $G = 16$ rollouts from the 24 rollouts of each scenario, and re-run the complete pipeline on the subset only, including re-clustering and re-estimating $g_x(p)$ and $m_x(p)$. This procedure is repeated 100 times per scenario, and the results are compared against the reference.

D.1.2 Evaluation Metrics

We report the following metrics:

- **Purity**: the fraction of members in each subset cluster that belong to its matched reference pattern, averaged over clusters, trials and scenarios. It measures whether the subset clustering recovers the same patterns as the reference.
- **MAE**: the mean absolute error between the efficiency score $g_x(p)$ estimated from the subset and that of the matched reference pattern.
- **Direction**: the fraction of trials in which the pattern assigned the smallest KL coefficient, i.e., the pattern with the highest estimated $g_x(p)$, matches that of the reference. Since $m_x(p)$ decreases monotonically with $g_x(p)$, this metric depends only on the ranking of $g_x(p)$ and is independent of the choice of α and γ.

D.1.3 Results

Clustering is stable. The purity reaches 0.94 at $G = 8$ and 0.93 at $G = 16$, indicating that the members of each pattern are consistently recovered from a small number of rollouts.

The rank of $g_x(p)$ is robust. Although the value of $g_x(p)$ is noisy, PSM mainly relies on the rank among patterns. The gap in $g_x(p)$ between patterns has a median of 0.57, well exceeding the noise level. As a result, the modulation direction is preserved in 97.6% of trials at $G = 8$ and 97.0% at $G = 16$, indicating limited impact of estimation noise on PSM.

D.2 Detail of Clustering Ablation

To investigate the impact of different clustering algorithms, we compare K-Means with an alternative scheme based on the Gaussian Mixture Model (GMM) with the Bayesian Information Criterion (BIC), which determines the number of clusters adaptively.

D.2.1 Experimental Setup

For the same scenarios sampled from the Bench2Drive training set used in our post-trainin, generate $G = 8$ rollouts for each scenario with ORION. Both clustering schemes operate on the same planning tokens and share the same preprocessing, in which PCA retains the principal components that explain 95% of the variance.

K-Means. We follow the setting of our main experiments with $K = 3$.

GMM+BIC. For each scenario, we fit GMMs with $K \in \{1, 2, 3\}$ components and select the one with the lowest BIC, defined as $\mathrm{BIC} = -2 \ln \hat{L} + k \ln n$, where $\hat{L}$ is the maximized likelihood, k the

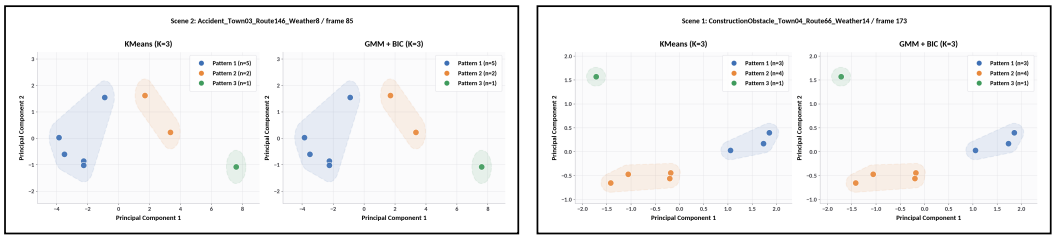


Figure 7: Clustering results of K-Means and GMM+BIC algorithms on the same example scenarios, showing highly consistent.

number of parameters, and n the number of samples. We adopt spherical covariance, which has the fewest parameters and is thus suitable for the small group size, and use three random initializations for each fit. In addition to hard assignments, GMM provides the posterior probability of each rollout belonging to each cluster, whose maximum we refer to as the assignment certainty.

For each scenario, we record the number of clusters selected by BIC, the Adjusted Rand Index (ARI) between the two partitions, the assignment certainty of GMM, and the KL coefficient assigned to each rollout by PSM under each partition. ARI measures the agreement between two partitions regardless of cluster labels, where 1 indicates identical partitions and 0 corresponds to random agreement.

D.2.2 Results

Table 7: Comparison between K-Means ($K = 3$) and GMM+BIC over same scenarios.

Statistic	Value
Scenarios where BIC selects $K = 3$	100%
Mean ARI	0.79
Scenarios with identical partitions (ARI $= 1$)	56%
Scenarios with $0.5 \leq$ ARI < 1	25%
Scenarios with ARI < 0.5	19%
Mean / minimum assignment certainty of GMM	0.999 / 0.966
Mean relative difference in KL coefficients	5.5%
Rollouts with relative difference $> 10\%$	13.3%
Rollouts with relative difference $> 50\%$	3.1%

Number of clusters. As shown in Table 7, BIC selects three clusters in all scenarios, consistent with the fixed choice of K-Means. This suggests that $K = 3$ reflects the intrinsic structure of the sampled planning tokens rather than an arbitrary setting.

Partition consistency. The two schemes produce identical partitions in most scenarios, with a mean ARI of 0.79. The disagreements mainly arise from a few boundary rollouts. Moreover, several scenarios with low ARI are nearly homogeneous in efficiency, e.g., all rollouts receive the same efficiency reward, or their speed ratios fall within a narrow range. In such scenarios, the estimated $g_x(p)$ is similar regardless of the partition, and the resulting modulation is barely affected.

Assignment certainty. The assignment certainty of GMM is at least 0.966 for all rollouts, indicating that the planning tokens form well-separated clusters. Hence, the hard assignment of K-Means does not introduce ambiguity for boundary rollouts in practice.

Impact on modulation. The KL coefficients assigned by PSM under the two partitions differ by only 5.5% on average relative to the base coefficient, and differ by more than 50% for only 3.1% of rollouts.

Overall, K-Means with $K = 3$ yields partitions and modulation consistent with the adaptive GMM+BIC scheme, while requiring only a single clustering per group instead of multiple EM fits over different values of K. We therefore adopt K-Means in PapoDrive for its efficiency.

Figure 7 compares the clustering results of K-Means and GMM+BIC on example scenarios, showing highly consistent partitions between the two algorithms.

E Proof of Proposition 1: Optimal Policy of RL Post-Training

Decompose the two terms in the optimization objective equation 3 into "two-stage processes".

The first term, "Expected Reward," is decomposed as follows:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x}),\boldsymbol{\tau}\sim\pi_{\text{planner}}(\cdot|\boldsymbol{x},p)}[r_{\text{safety}}(\boldsymbol{x},\boldsymbol{\tau})] &= \sum_p \sum_{\boldsymbol{\tau}} \pi_\theta(p|\boldsymbol{x})\pi_{\text{planner}}(\boldsymbol{\tau}|p,\boldsymbol{x})r_{\text{safety}}(\boldsymbol{x},\boldsymbol{\tau}) \\ &= \sum_p \pi_\theta(p|\boldsymbol{x}) \underbrace{\left(\sum_{\boldsymbol{\tau}} \pi_{\text{planner}}(\boldsymbol{\tau}|p,\boldsymbol{x})r_{\text{safety}}(\boldsymbol{x},\boldsymbol{\tau})\right)}_{s_x(p)} \\ &= \sum_p \pi_\theta(p|\boldsymbol{x}) s_x(p)\end{aligned} \tag{18}$$

where $s_x(p) = \mathbb{E}[r_{\text{safety}}(\boldsymbol{x},\boldsymbol{\tau}) \mid p]$ is the safety reward probability for pattern p given input $\boldsymbol{x}$.

The second term, "KL Divergence," is decomposed as follows:

$$\begin{aligned}\beta \cdot \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) &= \beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x}),\boldsymbol{\tau}\sim\pi_{\text{planner}}(\cdot|p,\boldsymbol{x})} \left[\ln \pi_\theta(\boldsymbol{\tau}|\boldsymbol{x}) - \ln \pi_{\text{ref}}(\boldsymbol{\tau}|\boldsymbol{x})\right] \\ &= \beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x}),\boldsymbol{\tau}\sim\pi_{\text{planner}}(\cdot|p,\boldsymbol{x})} \left[\ln\left(\frac{\pi_\theta(p|\boldsymbol{x})\cdot\pi_{\text{planner}}(\boldsymbol{\tau}|p,\boldsymbol{x})}{\pi_{\text{ref}}(p|\boldsymbol{x})\cdot\pi_{\text{planner}}(\boldsymbol{\tau}|p,\boldsymbol{x})}\right)\right]\end{aligned} \tag{19}$$

The planner's conditional distribution $\pi_{\text{planner}}(\boldsymbol{\tau}|p,\boldsymbol{x})$ remains approximately unchanged before and after fine-tuning. Therefore, the above equation simplifies to:

$$\beta \cdot \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) = \beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x})} \left[\ln\left(\frac{\pi_\theta(p|\boldsymbol{x})}{\pi_{\text{ref}}(p|\boldsymbol{x})}\right)\right] \tag{20}$$

Substituting equation 18 and equation 20 into the original objective equation 3, we obtain a simplified objective that depends only on the LLM's reasoning pattern distribution $\pi_\theta(p|\boldsymbol{x})$:

$$\mathcal{J}_{\text{RLVR}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x})} \left[s_x(p) - \beta \cdot \ln\left(\frac{\pi_\theta(p|\boldsymbol{x})}{\pi_{\text{ref}}(p|\boldsymbol{x})}\right)\right] \tag{21}$$

By moving $s_x(p)$ inside the logarithm and introducing a normalization constant $Z(\boldsymbol{x})$, we can rewrite the objective as:

$$\begin{aligned}\mathcal{J}_{\text{RLVR}}(\theta) &= -\beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},p\sim\pi_\theta(\cdot|\boldsymbol{x})} \left[\ln\left(\frac{\pi_\theta(p|\boldsymbol{x})}{\frac{1}{Z(\boldsymbol{x})}\cdot\exp(\frac{1}{\beta}s_x(p))\cdot\pi_{\text{ref}}(p|\boldsymbol{x})}\right)\right] + \beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ln Z(\boldsymbol{x})] \\ &= -\beta \cdot \mathbb{D}_{\text{KL}}\left(\pi_\theta(p|\boldsymbol{x}) \middle\| \frac{\exp(\frac{1}{\beta}s_x(p))\cdot\pi_{\text{ref}}(p|\boldsymbol{x})}{Z(\boldsymbol{x})}\right) + \beta \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ln Z(\boldsymbol{x})]\end{aligned} \tag{22}$$

where the partition function $Z(\boldsymbol{x})$ is defined as:

$$Z(\boldsymbol{x}) = \sum_{p\in\mathcal{P}} \exp\left(\frac{1}{\beta}s_x(p)\right)\pi_{\text{ref}}(p|\boldsymbol{x}) \tag{23}$$

The purpose of $Z(\boldsymbol{x})$ is to ensure the denominator constitutes a valid probability distribution.

Since β and $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ln Z(\boldsymbol{x})]$ are constants with respect to the policy π_θ, maximizing $\mathcal{J}_{\text{RLVR}}(\theta)$ is equivalent to minimizing the KL divergence term. The KL divergence reaches its minimum value of 0 when the two distributions are identical. Therefore, the optimal policy π_θ^* that maximizes the objective function is:

$$\pi_\theta^*(p \mid \boldsymbol{x}) = \frac{\exp\left(\frac{1}{\beta} s_x(p)\right) \pi_{\text{ref}}(p \mid \boldsymbol{x})}{Z(\boldsymbol{x})} \tag{24}$$

This completes the proof of Proposition 1.

F Proof of Proposition 2: Efficiency Drift

The efficiency drift is defined as:

$$\mathbb{E}_{\pi_\theta^*}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})] - \mathbb{E}_{\pi_{\text{ref}}}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})]. \tag{25}$$

Let the weighting factor be

$$w_x(p) := \exp(\frac{1}{\beta} s_x(p)), \tag{26}$$

By Proposition 1, the optimal policy satisfies

$$\pi_\theta^*(p \mid \boldsymbol{x}) = \frac{\exp\left(\frac{1}{\beta} s_x(p)\right) \pi_{\text{ref}}(p \mid \boldsymbol{x})}{Z(\boldsymbol{x})}. \tag{27}$$

Using the two-stage decomposition, the expected success under π_θ^* is:

$$\mathbb{E}_{\pi_\theta^*}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})] = \sum_{p\in\mathcal{P}} \pi_\theta(p \mid \boldsymbol{x}) \cdot g_x(p) = \frac{\sum_r \pi_{\text{ref}}(p \mid \boldsymbol{x})\, w_x(p)\, g_x(p)}{\mathbb{E}_{\pi_{\text{ref}}}[w_x(p)]} = \frac{\mathbb{E}_{\pi_{\text{ref}}}[\, g_x(p)\, w_x(p)\,]}{\mathbb{E}_{\pi_{\text{ref}}}[w_x(p)]}. \tag{28}$$

Now apply the covariance identity:

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \text{Cov}(X, Y), \tag{29}$$

with $X = g_x(p)$ and $Y = w_x(p)$ under policy π_{ref}, we get

$$\mathbb{E}_{\pi_\theta^*}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})] - \mathbb{E}_{\pi_{\text{ref}}}[r_{\text{efficiency}}(\boldsymbol{x}, \boldsymbol{\tau})] = \frac{\text{Cov}_{\pi_{\text{ref}}}(g_x(\mathcal{P}), w_x(\mathcal{P}))}{\mathbb{E}_{\pi_{\text{ref}}}[w_x(\mathcal{P})]}. \tag{30}$$

which is the statement of Proposition 2.

G Proofs for Pattern-level Step-size Modulation

Proof of Proposition 3. Write $\kappa(p) = \beta m_x(p) > 0$, $s(p) = s_x(p)$, and $\pi_0(p) = \pi_{\text{ref}}(p|x) > 0$. The pattern-level objective is $J(\pi) = \sum_p \pi(p)s(p) - \sum_p \kappa(p)\pi(p)\log\frac{\pi(p)}{\pi_0(p)}$ on the probability simplex.

Concavity. Each term $h_p(t) = \kappa(p)\, t \log(t/\pi_0(p))$ has $h_p''(t) = \kappa(p)/t > 0$. Hence J is strictly concave, and any stationary point of the Lagrangian in the interior of the simplex is the unique global maximizer.

Stationarity. Take the Lagrangian $\mathcal{L} = J(\pi) - \mu\big(\sum_p \pi(p) - 1\big)$. Setting $\partial\mathcal{L}/\partial\pi(p) = 0$ gives $s(p) - \kappa(p)\big(\log\frac{\pi(p)}{\pi_0(p)} + 1\big) - \mu = 0$, that is, $\pi(p) = \pi_0(p)\exp\big(\frac{s(p)-\mu}{\kappa(p)} - 1\big) > 0$. The point is interior, so the non-negativity constraints are inactive.

Existence and uniqueness of μ^*. Let $F(\mu) = \sum_p \pi_0(p)\exp\big(\frac{s(p)-\mu}{\kappa(p)} - 1\big)$. F is continuous and strictly decreasing, with $F \to \infty$ as $\mu \to -\infty$ and $F \to 0$ as $\mu \to \infty$. Hence $F(\mu) = 1$ has a unique root μ^*.

Bounds. At $\mu = \max_p \tilde{s}(p)$, every exponent $\frac{s(p)-\mu}{\kappa(p)} - 1 = \frac{\tilde{s}(p)-\mu}{\kappa(p)} \le 0$, so $F \le \sum_p \pi_0(p) = 1$. At $\mu = \min_p \tilde{s}(p)$, every exponent is ≥ 0, so $F \ge 1$. Since F is monotone, $\mu^* \in [\min_p \tilde{s}(p), \max_p \tilde{s}(p)]$.

Special case. If $m_x \equiv 1$, then $\pi^* \propto \pi_0 \exp(s/\beta)$, which recovers Proposition 1.

Proof of Proposition 4. From Eq. equation 10, $\log \frac{\pi^*(p)}{\pi_0(p)} = \frac{s(p)-\mu^*}{\kappa(p)} - 1 = \frac{\tilde{s}(p)-\mu^*}{\kappa(p)}$. Since $\kappa(p) > 0$, this is positive iff $\tilde{s}(p) > \mu^*$.

Now suppose q is up-weighted and $s(p) - s(q) > \beta(m_{\max} - m_{\min})$. Then

$$\tilde{s}(p) - \tilde{s}(q) = \big(s(p) - s(q)\big) - \beta\big(m_x(p) - m_x(q)\big) > \beta(m_{\max} - m_{\min}) - \beta(m_{\max} - m_{\min}) = 0,$$

so $\tilde{s}(p) > \tilde{s}(q) > \mu^*$, and p is up-weighted as well.

Finally, $m_{\max} - m_{\min} = (1 + \alpha) - \frac{1}{1+\gamma} = \alpha + \frac{\gamma}{1+\gamma}$.