Decoupling Representation Robustness and Behavioral Robustness in End-to-End Driving A Feature-Space Robustness Module for Actor–Critic Policies

1st Yuanchen Liu
Department of Computer Science
China University of Geosciences
Wuhan, China

2nd Dongcheng Li*
Department of Computer Science
California State Polytechnic University - Humboldt
California, USA

***Abstract*—Closed-loop adversarial training has emerged as an effective paradigm for improving the safety of end-to-end autonomous driving policies. However, existing approaches primarily operate at the environment or observation level, leaving the robustness of internal representations largely implicit and poorly understood.**

In this paper, we introduce the *Feature-Space Robustness Module* (FRM), a plug-in component that explicitly regularizes latent representations in actor–critic systems via worst-case bounded perturbations, without modifying environment dynamics.

We evaluate FRM on a closed-loop driving benchmark with 500 scenarios from the Closed-Loop Adversarial Training for Safe End-to-End Driving (CAT), averaged over five random seeds. Critic-side FRM reduces value sensitivity by over $6\times$, while joint actor–critic regularization lowers policy divergence by more than 75%. In closed-loop evaluation, FRM achieves comparable crash rates to CAT but with reduced route completion, indicating a trade-off between safety and task efficiency. These results reveal a partial decoupling between value robustness and behavioral robustness.

***Index Terms*—Representation Robustness, Behavioral Robustness, Actor–Critic Methods, Feature-Space Regularization, End-to-End Autonomous Driving.**

I. Introduction

End-to-end driving learns perception, decision-making, and control in a single model. While such integration enables high adaptability, it also makes policies sensitive to distributional shifts and internal perturbations. Existing closed-loop adversarial training methods focus on environment- or observation-level perturbations and typically do not directly regularize latent representations.

We investigate whether robustness to environment-level adversaries ensures stability of latent representations in actor–critic policies. Motivated by prior work in robust reinforcement learning [1]–[3], we find that adversarially trained policies remain sensitive to bounded perturbations in latent feature space. Such perturbations induce deviations in value estimates and policy outputs, indicating that robustness to environment-level disturbances does not ensure stability of internal representations. However, how robustness propagates within the actor–critic pipeline remains unclear.

We introduce the *Feature-Space Robustness Module* (FRM), a plug-in component that regularizes actor and critic networks through worst-case bounded perturbations in latent feature space. FRM operates independently of environment dynamics and enables independent regularization of actor and critic representations. It can be integrated into existing closed-loop adversarial training frameworks without modifying the environment or observation space.

Contributions. The main contributions of this work are summarized as follows:

- We examine whether closed-loop adversarial training stabilizes latent representations in actor–critic driving policies and show that bounded feature perturbations still destabilize them.
- We introduce a feature-space robustness formulation (FRM) that enforces stability of latent representations under bounded perturbations without modifying environment dynamics.
- We disentangle actor and critic regularization and show that critic-side regularization stabilizes value estimates, whereas actor-side regularization more directly changes closed-loop behavior.
- We find that more stable value estimates do not consistently yield better closed-loop behavior in our setting, suggesting that representation robustness and behavioral robustness can diverge.

The source code and experimental data are publicly available at: https://github.com/Lycpro/autonomous.git

II. Related Work

A. Closed-Loop Adversarial Training for Autonomous Driving

Closed-loop adversarial training improves safety by introducing active adversaries rather than relying on passive data. Methods like CAT [4] jointly optimize adversarial agents with the ego policy to reduce collision rates, while others

* Corresponding author

utilize adversarial scenario generation or adaptive traffic behaviors [5], [6]. Recent benchmarks, such as DriveDreamer [7] and SafeBench [8], further advance evaluation via learned world models and systematic robustness testing.

Unlike these environment-level approaches, we study robustness in the internal feature representations of actor–critic policies, which remains underexplored in prior closed-loop driving research.

B. Robust Reinforcement Learning

Robust RL aims to ensure reliability under uncertainty. Early approaches utilized min–max optimization over uncertain dynamics or domain randomization [1], [9], [10]. Subsequent works addressed robustness against adversarial perturbations in action or observation spaces [11], [12].

While recent studies have begun exploring representation-level uncertainties [12], [13], they typically treat robustness as a global objective. In contrast, we focus on *where* robustness is manufactured inside actor–critic systems and how it propagates to closed-loop behavior in safety-critical tasks.

C. Feature-Space and Representation-Level Adversarial Learning

Adversarial feature learning is well-established in computer vision for improving network robustness [14]–[16]. Techniques such as latent adversarial perturbations offer efficient regularization without the high cost of input-level training [17].

In reinforcement learning, however, most methods still operate on observations or actions, and the impact of feature-level perturbations on actor–critic dynamics remains underexplored. Existing studies have not systematically examined how perturbations in latent representations affect value estimation and policy behavior in closed-loop settings [17].

D. Actor–Critic Robustness and Stability

Actor–critic algorithms such as Twin Delayed Deep Deterministic Policy Gradient (TD3) [18] and Soft Actor-Critic (SAC) [19] are central to continuous control, yet critic instability can amplify errors and degrade performance. While prior work links representation stability to learning performance [13], the relationship between value robustness and closed-loop behavioral robustness remains insufficiently characterized.

III. Method

A. Preliminaries

We formulate end-to-end autonomous driving as a continuous-control reinforcement learning problem. At each timestep t, the agent observes a state s_t, executes a continuous action a_t, and receives a scalar reward r_t. The policy is trained using an off-policy actor–critic algorithm, where an actor network π_θ maps states to actions and a critic network Q_ϕ estimates state–action values.

Both the actor and critic operate on intermediate feature representations produced by a feature encoder $g(\cdot)$. We denote the encoded representation as:

$$f = g(s). \tag{1}$$

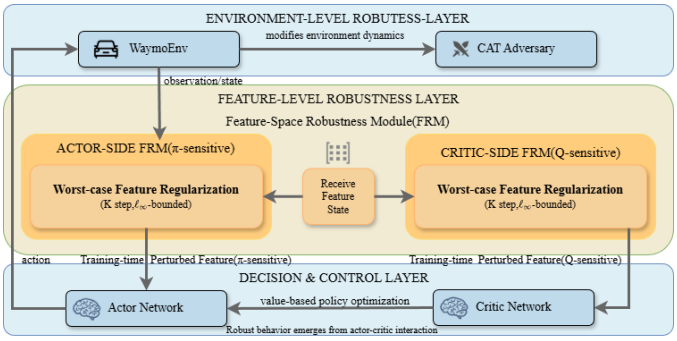

Fig. 1. Overview of the Feature-Space Robustness Module (FRM). Closed-loop adversarial training introduces external adversaries at the environment level. FRM instead regularizes internal representations by applying bounded perturbations in latent feature space. The module operates independently on the actor and critic during training, generating policy- and value-sensitive adversarial features. Closed-loop performance depends on how actor-side and critic-side regularization jointly affect policy updates and value estimates.

The actor and critic are then defined:

$$a = \pi_\theta(f), \qquad Q_\phi = Q_\phi(f, a). \tag{2}$$

Recent closed-loop adversarial training methods improve safety by introducing adversarial agents or perturbations at the environment level. In this work, we do not modify the environment-level adversary. Instead, we focus on explicitly enforcing robustness under bounded feature perturbations in the latent feature representations used by the actor–critic policy.

B. Feature-Space Robustness Formulation

We define feature-space robustness as the stability of policy outputs and value estimates under bounded perturbations applied directly to the latent feature representation. Given a feature vector f, we consider worst-case perturbations δ constrained by:

$$\|\delta\|_\infty \leq \epsilon, \tag{3}$$

and analyze the sensitivity of the policy and critic to the perturbed feature $f' = f + \delta$.

Unlike observation- or environment-level perturbations, feature-space perturbations leave sensory inputs and environment dynamics unchanged and directly test the sensitivity of the learned representation. This distinction is particularly important in actor–critic systems, where small perturbations to internal features can induce large deviations in value estimation and policy outputs, even when the environment remains unchanged.

The actor and critic are defined as follows. The actor outputs an action $a = \pi_\theta(f)$, while the critic estimates the state–action value $Q_\phi(f, a)$.

C. Feature-Space Robustness Module (FRM)

As illustrated in Fig. 1, we introduce FRM, a latent-space regularization module that applies bounded worst-case perturbations separately to actor and critic representations during

training without changing action execution or environment dynamics.

Given a module-specific objective $\mathcal{L}_{\text{FRM}}(f)$, FRM constructs adversarial features by solving the inner maximization problem:

$$f^{\text{adv}} = \arg \max_{\|\delta\|_\infty \le \epsilon} \mathcal{L}_{\text{FRM}}(f + \delta). \tag{4}$$

FRM applies adversarial training directly in latent feature space. Unlike input-level adversarial training, FRM performs this optimization directly in latent feature space, enabling targeted regularization of internal representations.

In practice, the inner maximization is approximated using projected gradient descent (PGD). Starting from an initial perturbation $\delta^{(0)} = 0$, FRM iteratively updates:

$$\delta^{(k+1)} = \Pi_{\|\delta\|_\infty \le \epsilon} \left(\delta^{(k)} + \alpha \nabla_f \mathcal{L}_{\text{FRM}}(f + \delta^{(k)}) \right), \tag{5}$$

where α is the step size and Π denotes projection onto the ℓ_∞-ball. The resulting adversarial feature is $f^{\text{adv}} = f + \delta^{(K)}$.

D. Critic-Side FRM

For the critic, FRM seeks feature perturbations that maximize the TD regression loss. Let

$$y = r + \gamma Q_{\phi^-}(f', \pi_\theta(f')) \tag{6}$$

denote the temporal-difference (TD) target, where ϕ^- represents the target critic parameters.

The critic-side FRM objective is defined as the TD regression loss:

$$\mathcal{L}^{\text{critic}}_{\text{FRM}}(f) = \left(Q_\phi(f, a) - y\right)^2. \tag{7}$$

FRM generates adversarial features by maximizing this loss under bounded perturbations:

$$f^{\text{adv}}_{\text{critic}} = \arg \max_{\|\delta\|_\infty \le \epsilon} \mathcal{L}^{\text{critic}}_{\text{FRM}}(f + \delta). \tag{8}$$

The critic parameters are then updated by minimizing the TD loss evaluated on the adversarial features:

$$\min_{\phi} \mathbb{E}\Big[\left(Q_\phi(f^{\text{adv}}_{\text{critic}}, a) - y\right)^2 \Big]. \tag{9}$$

This procedure regularizes the critic against worst-case feature perturbations and reduces the critic's sensitivity to local representation shifts.

E. Actor-Side FRM

For the actor, FRM seeks feature perturbations that minimize the critic-estimated value of the current policy. The actor-side FRM objective is defined as:

$$\mathcal{L}^{\text{actor}}_{\text{FRM}}(f) = -Q_\phi(f, \pi_\theta(f)). \tag{10}$$

Adversarial features are obtained by solving:

$$f^{\text{adv}}_{\text{actor}} = \arg \max_{\|\delta\|_\infty \le \epsilon} \mathcal{L}^{\text{actor}}_{\text{FRM}}(f + \delta). \tag{11}$$

The actor parameters are then updated by minimizing the adversarial policy objective:

$$\min_{\theta} \mathbb{E}\big[-Q_\phi\big(f^{\text{adv}}_{\text{actor}}, \pi_\theta(f^{\text{adv}}_{\text{actor}})\big) \big]. \tag{12}$$

The actor is optimized using adversarially perturbed features, while environment dynamics and executed actions remain unchanged.

F. Integration with Closed-Loop Adversarial Training

FRM is applied in addition to existing closed-loop adversarial training mechanisms. The environment-level adversary is unchanged.

During each training iteration:

1) The environment provides a transition (s, a, r, s').
2) The state is encoded into a feature representation $f = g(s)$.
3) FRM generates adversarial features for the actor, the critic, or both.
4) Actor and critic updates are performed using the corresponding FRM-regularized features.

This design regularizes latent representations without changing the environment-level adversary. Environment-level adversaries perturb external interactions, whereas FRM regularizes internal feature representations. This separation aligns with prior robust and distributionally robust formulations that distinguish uncertainty in dynamics from regularization of the decision function [2], [3], [20].

G. Actor–Critic Disentanglement

A key property of FRM is that it can regularize the actor and critic independently. This enables three configurations: actor-only FRM, critic-only FRM, and joint actor–critic FRM.

This design lets us compare actor-only, critic-only, and joint FRM to isolate their effects on value estimates and closed-loop behavior. As shown in our experiments, critic-side FRM plays a dominant role in stabilizing representation geometry, while actor-side FRM provides complementary effects on behavioral stability.

IV. Experimental Setup

A. Simulator and Scenarios

All experiments are conducted in a closed-loop autonomous driving simulator built on MetaDrive [21], following the experimental protocol of CAT [4]. The ego agent is trained to complete navigation tasks under dynamic and potentially adversarial traffic, where safety-critical events such as collisions may arise due to stochastic traffic behavior and policy decision errors.

The training and evaluation scenarios are primarily based on the 500 traffic scenarios provided by the CAT benchmark. During experiments, we observed that a small number of scenarios exhibit unstable execution behaviors in the simulator, leading to non-deterministic outcomes across runs. To ensure experimental consistency and reproducibility, we exclude these scenarios and replace them with two additional challenging traffic scenarios generated using ScenarioNet [22], following the same construction and filtering criteria as in CAT. All methods, including the CAT baseline and all FRM variants, are trained and evaluated on the same filtered and augmented scenario set.

The additional scenarios are generated from real-world traffic data in the Waymo Open Motion Dataset [23] and are selected to match the difficulty and interaction patterns of the original CAT scenarios.

B. Observations and Feature Representation

At each timestep, the observation is encoded into a fixed 101-dimensional latent feature vector, following the representation used in CAT. This latent feature serves as the shared input to both the actor and critic networks.

The proposed Feature-Space Robustness Module (FRM) operates directly on this shared latent representation during training. By perturbing internal features rather than raw observations or environment dynamics, FRM enables a focused analysis of representation-level robustness within the actor–critic pipeline under closed-loop driving interactions.

C. Baselines

We compare our method against the original CAT framework, which applies adversarial perturbations at the observation level during training.

To ensure a fair comparison, all methods share the same adversarial generator and training protocol as CAT. Our approach augments CAT with FRM, which introduces additional feature-space regularization without altering the original adversarial mechanism.

D. Training Details

All policies are trained using the same actor–critic architecture and optimization hyperparameters. FRM generates adversarial feature perturbations via projected gradient descent under an ℓ_∞ constraint, with perturbation budget $\epsilon = 0.05$, step size 0.01, and 3 iterations. Unless otherwise specified, FRM is applied to both actor and critic during training. All results are averaged over multiple evaluation episodes and five random seeds.

All experiments are conducted on a server equipped with an Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz and an NVIDIA RTX 4090 GPU.

V. Experimental Results

A. Research Questions

To systematically evaluate the proposed method, we organize our experiments around the following research questions:

RQ1: Does improved representation robustness translate to improved behavioral robustness in closed-loop settings?

RQ2: How does robustness regularization applied to the actor and critic individually affect closed-loop driving performance?

RQ3: Does feature-space robustness improve representation stability in actor–critic systems?

B. Main Results

To answer **RQ1**, we compare closed-loop performance with the CAT baseline (Fig. 2). Joint actor–critic FRM does not consistently reduce crash rates relative to CAT but avoids the large route-completion degradation observed with actor-only regularization. Critic-only FRM improves representation-level metrics but does not reduce crash rates.

Route completion decreases across all FRM variants, with the largest drop under actor-only regularization and the smallest degradation under joint regularization.

C. Actor–Critic Ablation Study

To answer **RQ2**, we evaluate actor-only, critic-only, and joint FRM (Table I).

Critic-side FRM produces the largest reductions in value sensitivity and gradient norm (Table II). However, these improvements do not reduce crash rates and are associated with decreased route completion.

Actor-only FRM provides limited robustness gains and substantially reduces route completion, indicating conservative policies. Joint actor–critic FRM achieves lower policy divergence while maintaining similar crash rates with smaller degradation in route completion.

Value stabilization alone is insufficient in this experimental setting, while combining it with policy regularization yields improved performance. [18], [24], [25].

D. Feature-Space Robustness Analysis

To answer **RQ3**, we quantify robustness under bounded perturbations $f' = f + \delta$ with $\|\delta\|_\infty \le \epsilon$.

a) Critic sensitivity: Value deviation is defined as:

$$\Delta_Q = \mathbb{E}\Big[\|Q_\phi(f,a) - Q_\phi(f',a)\|_2\Big]. \tag{13}$$

Critic-side FRM reduces $\|Q(f) - Q(f')\|$ by more than $6\times$ compared to the CAT baseline (Table II), with similar reductions in the 95% quantile. This indicates lower sensitivity of the critic to bounded feature perturbations.

b) Policy sensitivity: Policy divergence is defined as:

$$\Delta_\pi = \mathbb{E}\Big[\mathrm{KL}\big(\pi_\theta(\cdot|f) \,\|\, \pi_\theta(\cdot|f')\big)\Big]. \tag{14}$$

Actor-only FRM increases KL divergence, whereas joint FRM yields the lowest divergence among the compared variants (Table II). This is consistent with the degraded closed-loop performance observed under actor-only regularization.

c) Local smoothness: We define:

$$G_Q = \mathbb{E}\Big[\|\nabla_f Q_\phi(f,a)\|_2\Big]. \tag{15}$$

Critic-side FRM significantly reduces gradient norm, indicating improved local smoothness. Joint regularization maintains low critic sensitivity and lower policy divergence than actor-only regularization.

Overall, critic-side regularization primarily reduces value sensitivity and gradient norm, whereas actor-side regularization mainly affects policy divergence. In our experiments, lower value sensitivity did not consistently translate into lower crash rates.

E. Cross-Objective Feature Attacks

We evaluate generalization by constructing actor-targeted feature perturbations:

$$\max_{\|\delta\|_\infty \le \epsilon} \|\pi_\theta(f+\delta) - \pi_\theta(f)\|_2^2. \tag{16}$$

Fig. 3 reports action deviation and route-completion drop under varying ϵ. Both critic-only and joint FRM reduce local action sensitivity compared to CAT.

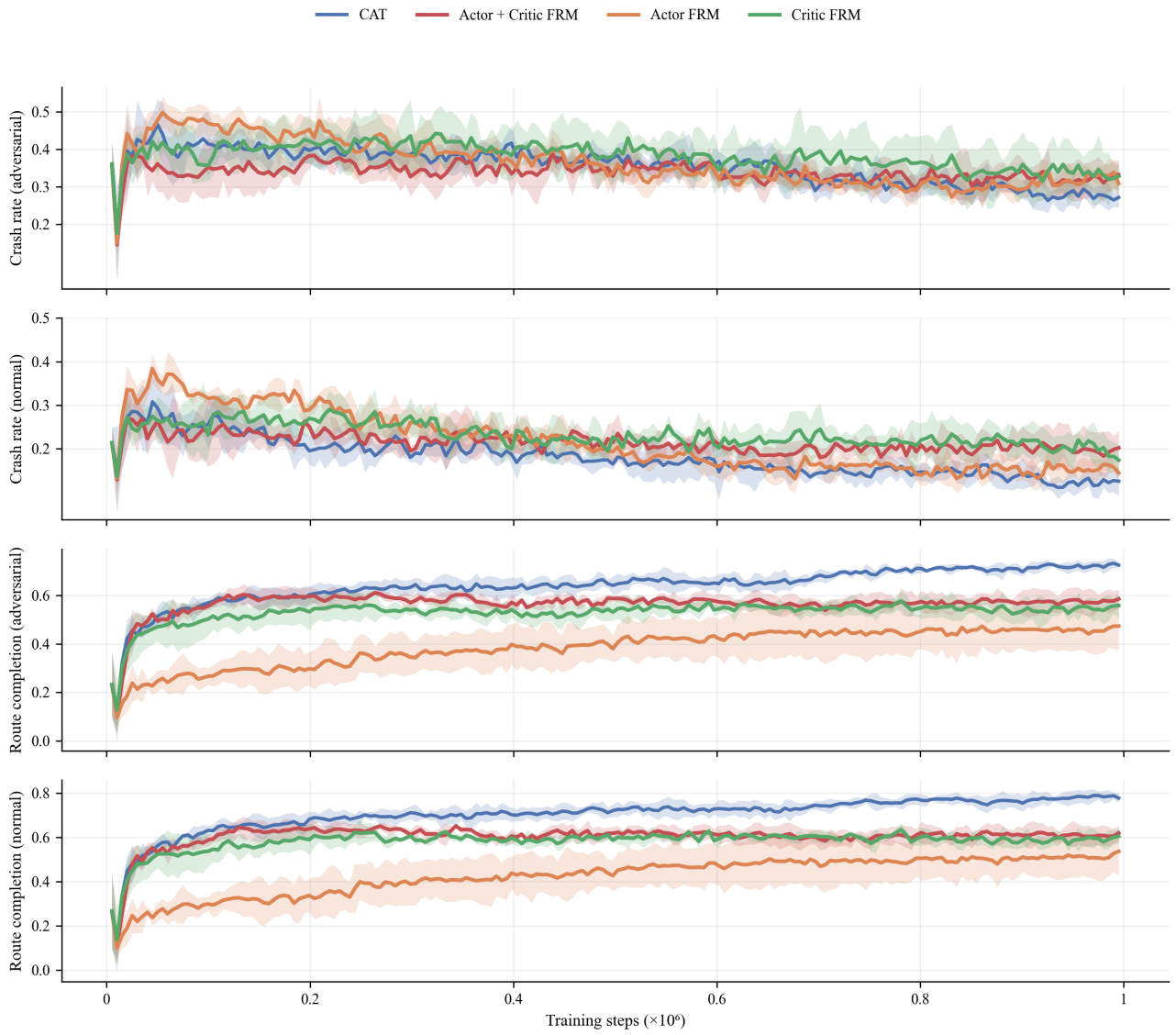

Fig. 2. Closed-loop training curves comparing the CAT baseline and FRM variants. The first two rows report crash rates under adversarial and normal evaluation settings, respectively. The last two rows report route completion rates under adversarial and normal evaluation settings. Solid lines show the mean metric value over five random seeds, and shaded regions denote standard deviation.

Critic-only FRM achieves the lowest action deviation but leads to larger drops in route completion. Joint FRM shows slightly higher local sensitivity but reduces performance degradation in closed-loop evaluation.

In these experiments, lower local action deviation did not consistently correspond to smaller route-completion drops. This further supports the observed decoupling between local robustness and closed-loop performance.

VI. Discussion

Improvements in value-level stability do not consistently translate into improvements in closed-loop performance in actor–critic policies. Critic-side FRM significantly reduces sensitivity to bounded feature perturbations (Table II), yet it does not reduce crash rates and often leads to decreased route completion (Fig. 2). Similar discrepancies have been observed in prior work, where local value improvements fail to translate into reliable behavior under distributional shift or adversarial perturbations [1], [26].

A. Why Feature-Space Robustness Matters

Prior work primarily enforces robustness at the observation or environment level [4], [27]. However, policies trained under such settings remain sensitive in latent space, as reflected by increased value deviation and policy divergence (Table II). Direct feature regularization reduces the sensitivity of value estimates and policy outputs to bounded latent perturbations and can complement environment-level adversarial training.

TABLE I

VALUES REPORT MEAN ± STANDARD DEVIATION OF PERFORMANCE DIFFERENCES RELATIVE TO THE CAT BASELINE. NEGATIVE VALUES INDICATE IMPROVEMENT FOR CRASH RATE (LOWER IS BETTER), WHILE ROUTE COMPLETION VALUES REFLECT PERFORMANCE DEGRADATION (LOWER IS WORSE). THE BEST RESULTS (LOWEST CRASH RATE AND SMALLEST DEGRADATION IN ROUTE COMPLETION) ARE HIGHLIGHTED IN BOLD.

Method	Crash Rate (Adv) ↓	Crash Rate (Norm) ↓	Route Completion (Adv) ↑	Route Completion (Norm) ↑
+ Actor FRM	$+0.006 \pm 0.016$	$+0.031 \pm 0.018$	$\mathbf{-0.243 \pm 0.074}$	$\mathbf{-0.287 \pm 0.085}$
+ Critic FRM	$+0.014 \pm 0.051$	$+0.041 \pm 0.025$	-0.102 ± 0.060	-0.103 ± 0.045
+ Actor + Critic FRM	$\mathbf{-0.025 \pm 0.021}$	$\mathbf{+0.025 \pm 0.016}$	-0.067 ± 0.047	-0.081 ± 0.050

TABLE II

FEATURE-LEVEL ROBUSTNESS METRICS UNDER BOUNDED FEATURE PERTURBATIONS ($\|\delta\|_\infty \leq \epsilon$). VALUES ARE REPORTED AS MEAN ± STANDARD DEVIATION OVER FIVE RANDOM SEEDS. LOWER IS BETTER FOR ALL METRICS.

Method	Mean $\|Q(f) - Q(f')\|_2 \downarrow$	95% Quantile ↓	Mean KL(π) ↓	Mean $\|\nabla_f Q\|_2 \downarrow$
CAT (baseline)	13.48 ± 0.08	23.05 ± 0.41	4.38 ± 3.71	0.085 ± 0.004
Actor-FRM	12.05 ± 0.62	20.41 ± 1.15	7.92 ± 1.24	0.070 ± 0.004
Critic-FRM	$\mathbf{2.11 \pm 0.17}$	$\mathbf{5.16 \pm 0.28}$	2.14 ± 0.72	$\mathbf{0.018 \pm 0.002}$
Actor + Critic FRM	2.49 ± 0.23	5.93 ± 0.38	$\mathbf{0.93 \pm 0.39}$	0.023 ± 0.003

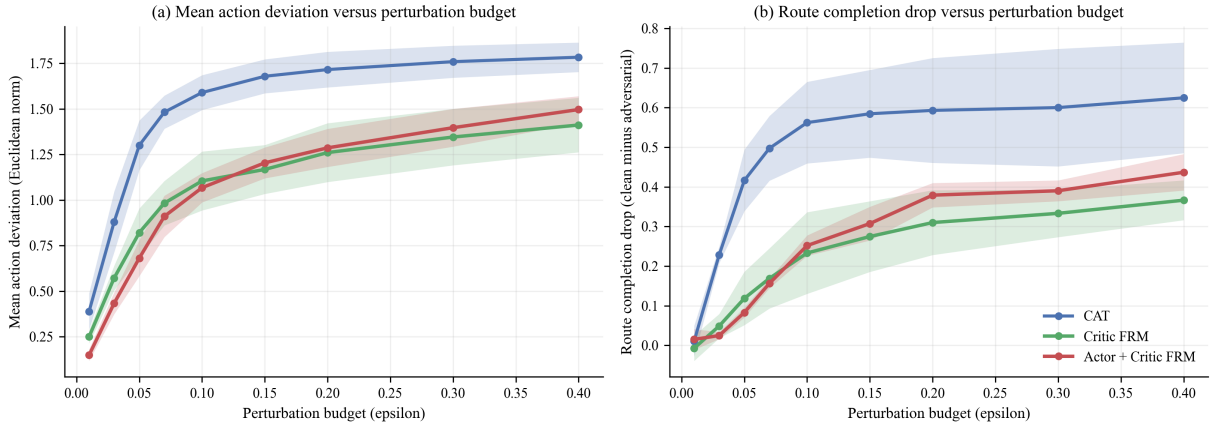

Fig. 3. Cross-objective feature attack evaluation. (**a**) Mean action deviation under critic-targeted feature perturbations with varying perturbation budgets ϵ. (**b**) Route completion drop under the same perturbations, measured relative to clean evaluation. Results are averaged over five random seeds, with shaded regions indicating one standard deviation.

B. Effect of Critic-Side Regularization

The ablation study (Table I) shows that critic-side FRM produces the largest reductions in value sensitivity and gradient norm (Table II), consistent with prior findings on stable value estimation [18], [28], [29]. However, this improvement does not reduce crash rates and is associated with decreased route completion (Fig. 2). Stabilizing the critic alone does not improve closed-loop safety metrics and is associated with degraded route completion, indicating that critic-side regularization is insufficient without policy-side regularization.

C. Representation Robustness vs. Behavioral Robustness

Feature-level metrics (Table II) measure local sensitivity to perturbations, while closed-loop metrics (Fig. 2) evaluate task-level performance (e.g., crash rate and route completion). Improvements in value deviation and policy divergence do not correspond to improvements in route completion or crash rate. Small perturbations in feature space can accumulate over time, leading to trajectory-level deviations in behavior.

D. Practical Implications

FRM integrates into existing actor–critic pipelines without modifying policy architectures or simulators [18], [19]. It is also compatible with environment-level adversarial training frameworks such as CAT [4]. Joint actor–critic regularization reduces policy divergence while maintaining similar crash rates with smaller degradation in route completion (Table I and Fig. 2).

E. Threats to Validity

The experiments are conducted on a single benchmark based on MetaDrive, and validation across additional simulators would strengthen generality. Moreover, feature-level metrics capture local sensitivity but do not fully explain long-horizon performance. Establishing a quantitative link between feature-level sensitivity metrics and long-horizon decision quality remains an open problem.

VII. Conclusion

We conclude that representation robustness and behavioral robustness are structurally misaligned in standard actor-critic policies; enforcing value stability alone fails to prevent compounding trajectory errors, necessitating joint regularization paradigms. These results indicate that representation robustness and behavioral robustness are not aligned, and that stable closed-loop performance requires joint regularization of actor and critic components.

Limitations and Future Work. First, our results indicate that representation-level stability and long-horizon behavioral performance are not captured by a single objective. One direction is to design training objectives that explicitly couple value sensitivity with closed-loop performance metrics. Second, while FRM bounds worst-case l_∞ latent perturbations, extending this regularization to structured, semantically meaningful feature shifts (e.g., simulated sensor degradation) remains a critical step toward real-world deployment. Third, existing feature-level metrics do not capture the relationship between local sensitivity and trajectory-level performance. Developing evaluation metrics that align feature robustness with closed-loop behavior is a key direction for future work.

References

[1] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2817–2826.

[2] G. N. Iyengar, "Robust dynamic programming," *Math. Oper. Res.*, vol. 30, no. 2, pp. 257–280, 2005.

[3] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Oper. Res.*, vol. 53, no. 5, pp. 780–798, 2005.

[4] L. Zhang, Z. Peng, Q. Li, and B. Zhou, "Cat: closed-loop adversarial training for safe end-to-end driving," in *Proc. Conf. Robot Learning*, 2023, pp. 2357–2372.

[5] A. A. Danso and U. Büker, "Automated generation of test scenarios for autonomous driving using llms," *Electronics*, vol. 14, no. 16, p. 3177, 2025.

[6] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 8271–8277.

[7] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "Drivedreamer: towards real-world-drive world models for autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 55–72.

[8] C. Xu, W. Ding, W. Lyu, Z. Liu, S. Wang, Y. He, H. Hu, D. Zhao, and B. Li, "Safebench: a benchmarking platform for safety evaluation of autonomous vehicles," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 25 667–25 682, 2022.

[9] J. Morimoto and K. Doya, "Robust reinforcement learning," *Neural Comput.*, vol. 17, no. 2, pp. 335–359, 2005.

[10] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "Epopt: learning robust neural network policies using model ensembles," 2016, arXiv:1610.01283. [Online]. Available: https://arxiv.org/abs/1610.01283.

[11] C. Tessler, Y. Efroni, and S. Mannor, "Action robust reinforcement learning and applications in continuous control," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6215–6224.

[12] Z. Liu, Z. Guo, Z. Cen, H. Zhang, J. Tan, B. Li, and D. Zhao, "On the robustness of safe reinforcement learning under observational perturbations," 2022, arXiv:2205.14691. [Online]. Available: https://arxiv.org/abs/2205.14691.

[13] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," 2020, arXiv:2007.05929. [Online]. Available: https://arxiv.org/abs/2007.05929.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, arXiv:1706.06083. [Online]. Available: https://arxiv.org/abs/1706.06083.

[15] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7472–7482.

[16] C. Xie, Y. Wu, L. van der Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 501–509.

[17] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[18] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[20] H. Xu and S. Mannor, "Distributionally robust markov decision processes," *Adv. Neural Inf. Process. Syst.*, vol. 23, 2010.

[21] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3461–3475, 2022.

[22] Q. Li, Z. Peng, L. Feng, Z. Liu, C. Duan, W. Mo, and B. Zhou, "Scenarionet: open-source platform for large-scale traffic scenario simulation and modeling," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 3894–3920, 2023.

[23] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, and Y. Zhou, "Large-scale interactive motion forecasting for autonomous driving: the waymo open motion dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9710–9719.

[24] H. Zhang, H. Chen, D. Boning, and C.-J. Hsieh, "Robust reinforcement learning on state observations with learned optimal adversary," 2021, arXiv:2101.08452. [Online]. Available: https://arxiv.org/abs/2101.08452.

[25] Z. Li, C. Hu, S. E. Li, J. Cheng, and Y. Wang, "Robust safe reinforcement learning under adversarial disturbances," in *Proc. IEEE Conf. Decis. Control*, 2023, pp. 334–341.

[26] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," 2017, arXiv:1712.03632. [Online]. Available: https://arxiv.org/abs/1712.03632.

[27] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, "Deep learning-based autonomous driving systems: a survey of attacks and defenses," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 7897–7912, 2021.

[28] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 12, 1999.

[29] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.