Beta-Distributed Proximal Policy Optimization for Autonomous Driving with Grad-CAM Explainability

Ernesto Pedrazzini[1,*], Valentina Zangirolami[1], Sonia Migliorati[1], and Matteo Borrotti[1]

Università degli Studi di Milano-Bicocca, Milan, Italy
[*]e.pedrazzini4@campus.unimib.it

Abstract. This paper investigates the impact of action sampling distributions on the efficiency and interpretability of Proximal Policy Optimization (PPO) in autonomous driving tasks. We compare PPO with Gaussian, Truncated Normal and Beta sampling in the photorealistic AirSim Neighborhood environment. Our experiments demonstrate that the Beta distribution significantly improves learning speed, with the Beta-based agent achieving task completion in fewer timesteps. Moreover, Grad-CAM visualization reveals that Beta-trained policies develop more interpretable decision-making strategies, with a clear focus on relevant obstacles rather than exhibiting diffuse attention patterns. These findings suggest that the Beta distribution is preferable for PPO-based agents in bounded-action spaces, particularly in safety-critical domains that require both sample efficiency and explainability.

Keywords: Reinforcement Learning · Proximal Policy Optimization · AirSim · Beta Distribution · Explainable Ai · GradCAM.

1 Introduction

Learning optimal policies for autonomous vehicles requires processing high-dimensional visual inputs via Deep Reinforcement Learning (DRL). While early DRL successes focused on discrete action spaces, autonomous driving requires continuous control. Discretizing continuous actions leads to the curse of dimensionality [5], motivating the use of policy gradient methods. Among them, Proximal Policy Optimization (PPO) [11] has become widely adopted due to its empirical stability and practical simplicity [15]. A key design choice in PPO for continuous control is the action sampling distribution. In self-driving tasks, actions such as steering are inherently bounded (e.g., in $[-1, +1]$), yet the standard Gaussian distribution has unbounded support, forcing action clipping and introducing bias in gradient estimates. Two principled alternatives are the Truncated Normal [4], which restricts support to the valid interval, and the Beta distribution [4], intrinsically defined over a finite interval with flexible shape and no need for clipping. Beyond performance, DRL methods suffer from limited interpretability a critical

concern in safety-sensitive domains. To investigate how the choice of distribution affects the learned perception–action mechanism, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [12] not only as a qualitative visualization tool, but as the basis for a quantitative metric measuring how well each policy concentrates attention on semantically meaningful visual features.

In summary,this paper compares PPO under Gaussian, Truncated Normal, and Beta sampling in the photorealistic AirSim Neighborhood simulator [13], evaluating both sample efficiency and interpretability.

The remainder of the paper is organized as follows. Section 2 illustrates the main related works. Section 3 reviews the necessary background on reinforcement learning and policy gradient methods, and introduces the Grad-CAM visualization technique. Section 4 describes the experimental setup, including the neural network architecture. Section 5 presents the experimental results, comparing learning performance and sample efficiency under Gaussian, Truncated Normal, and Beta policies, along with the corresponding interpretability analysis. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Related literature

The growing complexity of deep learning models has motivated increasing interest in Explainable Artificial Intelligence (XAI), a field concerned with developing methods that make the behavior of machine learning systems interpretable and transparent. A comprehensive taxonomy of XAI concepts, techniques, and challenges is provided by [1], who highlight the fundamental tension between predictive accuracy and model interpretability, arguing that explainability is a prerequisite for the responsible deployment of AI in high-stakes domains. Among the most widely adopted post-hoc, model-agnostic explanation techniques are LIME [8] and SHAP [6], which provide local and global feature-level explanations respectively. Autonomous driving represents a particularly compelling domain for XAI, given the safety-critical nature of its decisions. [2] provide a comprehensive overview of XAI approaches for autonomous vehicles, outlining future research directions for transparent end-to-end driving systems. Since our policy network is a Convolutional Neural Network (CNN) with direct access to its internal activations, we opt for Grad-CAM [12] over perturbation-based approaches, as it produces spatially localized heatmaps more efficiently. A related use of gradient-based saliency maps in adversarial DRL for autonomous driving is explored in [19].

PPO has been successfully applied to autonomous driving in photorealistic simulated environments [9]. Replacing the Gaussian with the Beta distribution has been shown to improve sample efficiency in bounded action spaces [3], a finding confirmed across OpenAI Gym benchmarks [7]. Truncated Normal distributions offer a complementary approach to handling action constraints while preserving Gaussian symmetry [17]. This work extends these comparisons to a photorealistic 3D driving simulator and introduces an explainability-oriented evaluation via Grad-CAM, addressing a dimension overlooked in prior work.

3 Problem formulation

We consider Reinforcement Learning (RL) problems [18] in which data are generated by an infinite-horizon Markov decision process (MDP). Initial states are drawn from a distribution ρ, while state transitions follow a Markovian transition kernel $\mathcal{T}(s_{t+1}|s_t, a_t)$. The system is characterized by a continuous action space $\mathcal{A}$, the state space $\mathcal{S}$ and a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Under this framework, rewards are conditionally independent given the state–action pairs [14]. Actions are generated according to a stochastic behavior policy $a \sim \pi_\theta(a|s)$. The main goal of RL is to find an optimal policy π^* that maximizes the expected cumulative discounted reward $\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right]$, where $\gamma \in [0, 1)$ is the discount factor and $\tau = (s_0, a_0, r_0, \ldots)$ denotes a trajectory induced by the policy.

RL can be formulated in terms of value or Q-value functions that encode the expected discounted return induced by a policy. For a policy π, the associated state-value function is $V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t \mid s_0 = s\right]$. Under standard assumptions, finding the optimal value functions $V^* = \max_\pi V^\pi(s)$ leads to the optimal policy π^*. The advantage function $A^\pi(s, a)$ measures how much better it is to take action a in state s compared to the average action under policy π, and plays a central role in modern policy optimization methods.

3.1 Proximal Policy Optimization

Policy gradient methods directly parameterize a stochastic policy $\pi_\theta(a \mid s)$ and optimize the expected return $J(\theta) = V^{\pi_\theta}(s)$ via gradient ascent:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)A^{\pi_\theta}(s_t, a_t)\right]. \tag{1}$$

Vanilla policy gradient methods suffer from high variance and instability, motivating trust region approaches such as PPO [11]. PPO controls policy updates through a clipped surrogate objective. Defining $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ as the probability ratio between the updated and previous policy, the objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right], \tag{2}$$

where ϵ controls the clipping range. PPO further uses Generalized Advantage Estimation (GAE) [10] as a low-variance advantage estimator, interpolating between Monte-Carlo and temporal difference methods via a parameter $\lambda \in [0, 1]$: $A_t^{\text{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}$, where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the temporal difference error.

In continuous control tasks, the PPO policy is modeled as a parametric probability distribution π over actions. Consequently, the choice of this distribution plays a pivotal role in the learning dynamics and policy stability.

3.2 Gaussian Bias

When using Gaussian distributions for action sampling in bounded continuous action spaces, a fundamental limitation arises. Consider an action space confined to a bounded interval such as $[-1, 1]$. The Gaussian distribution is unbounded by nature, assigning non-zero probability to values outside the valid action interval. In practice, this issue is addressed through action clipping, but this introduces a bias in the action distribution, as illustrated in Figure 1, leading to suboptimal learning dynamics.

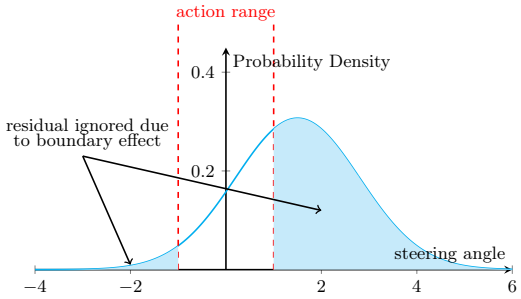

Fig. 1. Illustration of Gaussian bias in bounded action spaces. The shaded regions represent probability mass that falls outside the valid action range $[-1, 1]$.

Truncated Normal Distribution An alternative approach to address the boundary bias consists in explicitly truncating the Gaussian distribution to the valid action interval. Let $X \sim \mathcal{N}(\mu, \sigma^2)$. The truncated Normal distribution on $[a, b]$ is defined by conditioning X to lie within the bounded interval. Its probability density function is given by

$$f(x; \mu, \sigma, a, b) = \frac{\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}, \quad x \in [a, b] \tag{3}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard Normal probability density function and cumulative distribution function, respectively [4]. The truncated Normal has support strictly confined to $[a, b]$, ensuring consistency between the sampling distribution and the executed actions. In our bounded steering setting, we consider $a = -1$ and $b = 1$.

Beta Distribution While the Truncated Normal preserves the Gaussian structure under bounded support, an alternative is to adopt a distribution that is intrinsically defined over a finite interval with a flexible and smoother shape.

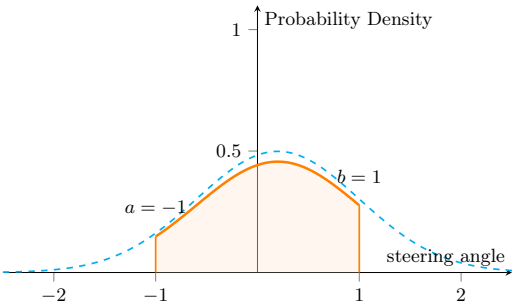

Fig. 2. Visualization of the Truncated Normal distribution over the interval $[-1, 1]$. The orange curve represents the renormalized density confined to the admissible action range, while the dashed cyan curve shows the original unbounded Gaussian.

The Beta distribution has the following probability density function:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{4}$$

where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ is the Beta function, and $\alpha, \beta > 0$ are shape parameters, while $\text{Beta}(\alpha, \beta)$ denotes the Beta distribution. This distribution, as illustrated in Figure 3, is natively defined on the interval $[0, 1]$, but can be easily rescaled to any bounded range, such as $[-1, 1]$, through a linear transformation.

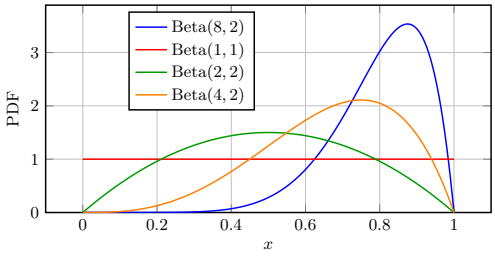

Fig. 3. Examples of Beta distributions for different α and β.

3.3 Grad-CAM

To address the explainability challenge, we employ Grad-CAM [12], which offers a more direct visualization than perturbation-based approaches such as LIME [8] and SHAP [6], specifically designed for CNNs. Grad-CAM computes gradient-based importance weights α_k^c for each feature map k in the final convolutional layer with respect to the target output c. These weights are calculated by global

average pooling of the gradients over the spatial dimensions: $\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$, where y^c is the output score for class (or action) c, A_{ij}^k represents the activation at spatial location (i, j) in feature map k, and Z is the number of spatial locations. The final Grad-CAM heatmap is then computed as:

$$L_{\text{Grad-CAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right). \tag{5}$$

The ReLU operation retains only features that positively contribute to the target output. By overlaying these heatmaps on the input images, we can visually identify which regions the policy network focuses on when selecting steering actions, providing interpretable insights into the learned driving behaviors.

To move beyond purely qualitative visualization, we propose a quantitative metric to compare the interpretability of policies trained under different action distributions. Given a Grad-CAM heatmap produced for a specific input frame, we define the *weighted spatial deviance* δ_w of the activations with respect to the known position of the relevant object in the scene. The heatmap activations are normalized to sum to 1 and used as probability weights, allowing us to compute the deviance of pixel positions with respect to the object's center:

$$\delta_w = \sum_{i,j} w_{ij}\, d\big((i,j),\,(i_{\text{obj}}, j_{\text{obj}})\big)^2, \quad w_{ij} = \frac{a_{ij}}{\sum_{i,j} a_{ij}} \tag{6}$$

where i and j are the row and column indices of the heatmap, $(i_{\text{obj}}, j_{\text{obj}})$ is the pixel position of the object's center, a_{ij} is the Grad-CAM activation value at pixel (i, j), and $d\big((i,j),\,(i_{\text{obj}}, j_{\text{obj}})\big)$ is a distance function measuring the displacement between a pixel and the object's center.

4 Experimental Setup

We tested our algorithms in a self-driving scenario using AirSim Neighborhood (AirSimNH) [13], a realistic driving simulator, in which the agent controls the steering angle of a vehicle, represented as a scalar value in the range $[-1, 1]$. We compare PPO with Gaussian, Truncated Normal, and Beta action distributions. The models are trained from scratch with identical hyperparameters and network architectures, differing only in the distribution used for action sampling. Figure 4 shows representative views of the two experimental scenarios used in this work.

4.1 Neural Network Architecture

The policy is parameterized by a neural network trained using the PPO objective. Since observations consist of raw visual inputs from the vehicle's front-facing camera, the network employs a CNN feature extractor processing $50 \times 50 \times 3$ RGB images, whose architecture is detailed in Table 1.

The final 512-dimensional feature vector is then fed into a distribution-specific output layer, whose number of neurons varies according to the chosen

Fig. 4. AirSim Neighborhood simulation environment. Left: obstacle avoidance scenario. Right: left-turn navigation scenario.

Table 1. CNN feature extractor architecture. Input: RGB image $(3, 50, 50)$.

Layer	Filters	Kernel/Stride	Output Shape
Conv2D + BN + ReLU	32	8×8, s=2	$(32, 22, 22)$
Conv2D + BN + ReLU	64	4×4, s=2	$(64, 10, 10)$
Conv2D + BN + ReLU	64	3×3, s=1	$(64, 8, 8)$
Flatten	-	-	(4096)
Linear + BN + ReLU	-	-	(512)

policy distribution: the Gaussian and Truncated Normal heads each output two parameters, μ and $\log \sigma$; the Beta head outputs the two shape parameters α and β, both constrained to be greater than 1 to ensure a unimodal distribution amenable to policy gradient optimization. All other network weights are shared across the compared distributions. While deeper or wider architectures may offer further gains, the chosen design ensures a fair and computationally tractable comparison across all three distribution variants; an investigation of architecture sensitivity is left as future work.

In this context, monitoring the learning performance of the algorithm is essential to extract meaningful insights regarding model quality and to enable principled comparisons across different policy parameterizations. To this end, we follow the approach recommended by Stable-Baselines3 [16], performing periodic deterministic evaluations throughout training: in those episodes, rather than sampling from the stochastic policy, actions are selected as the mode of the policy distribution, providing a more stable and reproducible performance estimate. All policy variants are trained using PPO with consistent hyperparameters to ensure a fair comparison: learning rate 3×10^{-4}, batch size 128, discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, clip range $\epsilon = 0.10$, and maximum gradient norm 0.5 over 12,288 total timesteps. These values follow established defaults from the PPO literature [11] and the Stable-Baselines3 guidelines [16], which have been shown to provide reliable convergence across a range of continuous control tasks. We evaluate every 1,024 timesteps in deterministic mode (actions selected via mode). Each evaluation consists of 15 episodes to provide

statistically meaningful performance estimates. All experiments are conducted on a Tesla M60 GPU using the AirSim Neighborhood environment.

4.2 Reward Function Design

We employed a distance-based reward function specifically designed to emphasize goal-reaching behavior. At each timestep t, we measure the Euclidean distance d_t between the vehicle's current position and the target. The reward r_t is computed as:

$$r_t = \begin{cases} c \cdot (d_{t-1} - d_t) & \text{if } d_t < d_{t-1} \\ k \cdot (d_{t-1} - d_t) & \text{if } d_t \geq d_{t-1} \end{cases} \tag{7}$$

where $c > 0$ rewards progress toward the goal and $k > 0$ penalizes moving away from it. Episodes terminate upon collision (penalizing the cumulative reward by a factor $1/d$), upon reaching the target (bonus reward of x units), or after exceeding a 13-second time limit. The vehicle's throttle and brake are automatically regulated to maintain a constant velocity of 5 m/s, allowing the agent to focus exclusively on learning the steering behavior. In our experiments, we set $c = 5$, $k = 5$, $d = 6$, and $x = 100$.

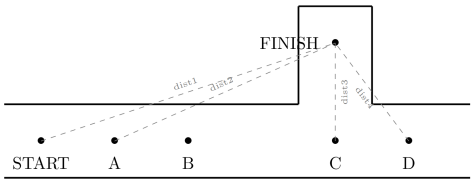

Fig. 5. Reward function scheme. The agent's trajectory gives a positive reward for going from A to B, and a negative one for going from C to D.

5 Results

In this section, we present the results of our comparative study between PPO with Gaussian, Truncated Normal and with Beta sampling on two different tasks in the AirSim Neighborhood environment: (i) **obstacle avoidance**: a scenario where the car must navigate around a central pole to reach a target 28 meters ahead; (ii) **left turn navigation**: a task in which the agent had to learn to turn left so as not to move away from the target point or collide with the fence. This focused approach allows the agent to learn specific, decomposed elementary maneuvers. The complete collision-free trajectory requires approximately 10 seconds and 30 meters to traverse in both cases. To conduct a controlled

comparison, for each task, training is performed from a single starting point in the environment (with added noise). This experimental design enables us to: (i) verify whether the Beta distribution provides the sample efficiency advantages predicted by [3], [7] in this complex visual domain, and (ii) compare the explainability of the learned policies using Grad-CAM.

5.1 Learning Performance and Sample Efficiency

In our experiments an episode was considered successfully completed when the vehicle reached the target point without any collisions. Performance is measured using three metrics computed over the 15 deterministic evaluation episodes run every 1,024 timesteps: the Average Reward and its Standard Deviation, and the average episode duration before a collision or timeout. The Collision Free Rate (CFR) is instead computed globally over all evaluation episodes throughout training, as the percentage of collision-free episodes out of the total.

The cumulative reward of the obstacle avoidance experiment obtained during these evaluation phases is shown in Figure 6. The results reveal that all models eventually succeed in completing the task when given sufficient training time. However, a crucial difference emerges in the sample efficiency. The Beta distribution-based model achieves significantly higher performance after 6144 timesteps, whereas the Truncated Normal model shows this transition at 7168 timesteps and the Gaussian model at 8192 timesteps.

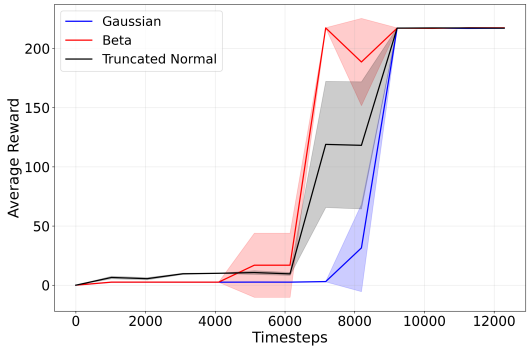

Fig. 6. Reward comparison for obstacle avoidance task. Red line: Beta model average reward; black line: Truncated Normal average reward; blue line: Gaussian model average reward during evaluation. Shaded regions represent 95% confidence intervals.

Table 2 summarizes the final performance metrics for both models on the obstacle avoidance task. The Beta distribution demonstrates a higher CFR compared to the Gaussian's and the Truncated Normal's, confirming that the Beta distribution's native boundedness provides tangible advantages in sample efficiency for this bounded control problem.

Table 2. Performance comparison on obstacle avoidance task.

Model	Avg Reward	Std Dev	CFR	Avg Time (s)
Gaussian	217.14	0.47	34.44%	8.16
Truncated Normal	217.19	0.50	42.22%	8.54
Beta	217.47	0.66	50%	8.71

The second experiment focused on the left-turn navigation task, in which the agent must learn to execute a controlled left turn to reach a target point on a perpendicular street. This task tests the agent's ability to learn smooth, goal-directed steering maneuvers in the absence of explicit obstacles. Following the same evaluation protocol, Figure 7 shows the cumulative reward progression for all models during deterministic evaluation. The results for the left-turn task show trends similar to those in the obstacle-avoidance scenario, with the Beta-based model again demonstrating superior sample efficiency.

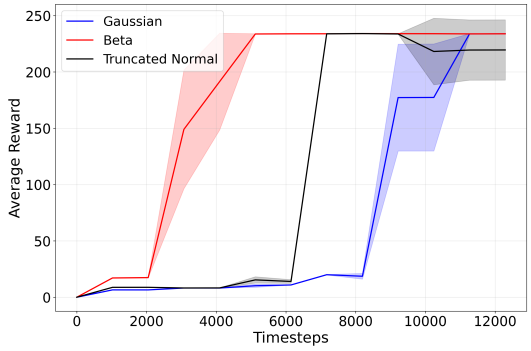

Fig. 7. Reward comparison for left turn navigation task. Red line: Beta model average reward; black line: Truncated Normal model average reward; blue line: Gaussian model average reward during evaluation. Shaded regions represent 95% confidence intervals.

Table 3 provides a quantitative summary of the performance metrics for the left turn task, confirming that the Beta distribution maintains its advantages in this different driving scenario.

Table 3. Performance comparison on the left turn navigation task.

Model	Avg Reward	Std Dev	CFR	Avg Time (s)
Gaussian	233.88	0.49	28.89%	9.01
Truncated Normal	219.52	52.86	48.33%	8.65
Beta	233.87	0.57	78.33%	9.07

5.2 Explainability Analysis with Grad-CAM

To understand the qualitative and quantitative differences in the learned policies, we applied Grad-CAM visualization to all the trained models, on a frame where the pole constitutes the primary semantically relevant feature for the steering decision.

Figure 8 shows representative Grad-CAM activation maps for all three models on the same input frame. The Gaussian-trained model's activations form a spatially clustered pattern, but one that is displaced from the pole. The Truncated Normal model exhibits even more diffuse attention, with activations scattered across the image and not clearly associated only with the obstacle. In contrast, the Beta-trained model sharply concentrates its activations on the pole, producing heatmaps that align intuitively with what a human expert would identify as the decisive visual feature for obstacle avoidance.

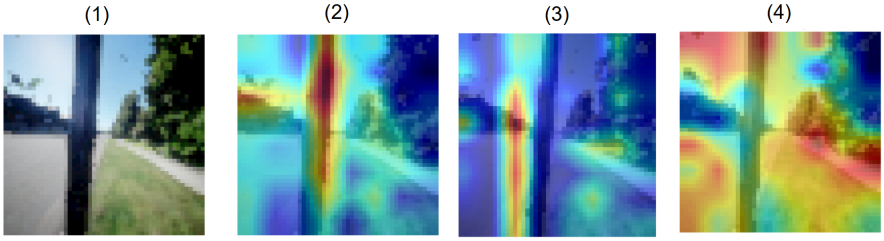

Fig. 8. Grad-CAM activation maps. From left to right: (1) original input frame with the pole obstacle visible at the centre-left of the image; (2) Beta model's activations; (3) Gaussian model's activations; (4) Truncated Normal model's activations.

Since the pole's elongated shape is spatially discriminative only along the horizontal axis, the deviance is computed with respect to the x-axis, with x_{obj} set to the known horizontal position of the pole's center in each frame. Specifically, on the single illustrative frame, the Beta model yields $\delta_w = 167.54$, the Gaussian $\delta_w = 186.67$, and the Truncated Normal $\delta_w = 194.42$.

To move beyond this single illustrative example and provide a statistically robust assessment, we computed the weighted spatial deviance δ_w across a sample of 10 distinct input frames, each capturing the pole obstacle from different viewpoints and distances encountered during the evaluation episodes. The broader 10-frame analysis, summarized in Figure 9, confirms the qualitative observations from the single illustrative example. Across all frames, the Beta-trained model consistently achieves the lowest weighted spatial deviance, while the Gaussian and Truncated Normal models show comparably higher values. The boxplot confirms that the Beta-trained model achieves systematically lower weighted spatial deviance than both alternatives, with a clearly separated median and reduced variance, indicating more consistent and focused attention across diverse viewpoints and distances.

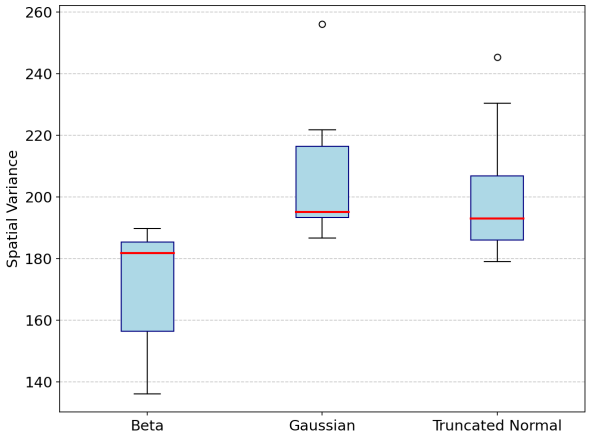

Fig. 9. Distribution of weighted spatial deviance δ_w across 10 evaluation frames for each model. Lower values indicate better concentration of attention around the obstacle.

These results suggest that the Beta distribution can not only lead to faster learning, but also to more interpretable decision-making strategies that align better with human intuition about which visual features should drive obstacle avoidance behavior. A policy whose attention concentrates on semantically meaningful regions is likely to be more robust and trustworthy in deployment scenarios, making this explainability dimension particularly relevant for safety-critical autonomous driving applications.

6 Conclusions and Limitations

The comparative study presented in this paper demonstrates that the choice of action sampling distribution has significant implications for both the efficiency and interpretability of deep reinforcement learning agents in autonomous driving tasks. Our experimental results in the AirSim Neighborhood environment confirm that PPO with Beta distribution sampling achieves substantially improved sample efficiency compared to the standard Gaussian implementation and the Truncated Normal one. These findings extend previous observations in simpler control environments [3], [7] to the more complex domain of vision-based autonomous driving, demonstrating that the benefits of Beta sampling are robust across different levels of task complexity. Beyond sample efficiency, our Grad-CAM analysis shows that the Beta-trained agent develops more interpretable strategies, with activation maps focused on the obstacle rather than diffuse patterns misaligned with human intuition. This is particularly valuable in safety-critical applications where validating the agent's decision process is essential.

Limitations. Several limitations of the current study should be acknowledged. First, regarding the *action space*: in this work the agent controls only the steering angle, a single scalar in $[-1, 1]$, while throttle and braking are handled by a predefined constant-velocity controller. This simplification allows a focused evaluation of the distribution choice, but real-world driving requires simultaneous optimization over a multi-dimensional action space; it is not guaranteed that the advantages of the Beta distribution scale uniformly to higher-dimensional or asymmetric action spaces. Second, the *network architecture* employed here is intentionally simple and may not capture the full perceptual complexity of modern autonomous driving stacks; deeper or multi-modal architectures could alter the relative advantage between distributions. Third, the *hyperparameters* were set to well-established PPO defaults and kept fixed across all variants; a broader hyperparameter search or ablation study may reveal settings where the Gaussian or Truncated Normal distributions perform more competitively. Finally, experiments are conducted in a single simulated environment with limited scene diversity, and generalization to more complex or real-world scenarios remains to be validated.

Future work. Future work should investigate the underlying theoretical mechanisms that explain why Beta sampling leads to both faster convergence and more interpretable attention patterns, as understanding these causal factors could guide the principled design of sampling distributions for other problems. A systematic hyperparameter sensitivity analysis and comparison across different network architectures would strengthen the empirical claims. Furthermore, it could be interesting to investigate whether these advantages extend to other policy gradient methods beyond PPO, such as Actor-Critic algorithms (e.g., SAC), and to multi-dimensional continuous action spaces, to establish the general validity of Beta distributions for improved performance across different algorithmic structures.

Acknowledgements The authors greatly acknowledge the DEMS Data Science Lab of the University of Milano-Bicocca.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
2. Atakishiyev, S., Salameh, M., Yao, H., Goebel, R.: Explainable Artificial Intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. IEEE Access **12**, 101603–101625 (2024). https://doi.org/10.1109/ACCESS.2024.3431437
3. Chou, P.W., Maturana, D., Scherer, S.: Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In: International conference on machine learning. pp. 834–843. PMLR (2017)

4. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, Volumes 1–2. John Wiley & Sons (1994)
5. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
6. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
7. Petrazzini, I.G., Antonelo, E.A.: Proximal policy optimization with continuous bounded action space via the beta distribution. In: 2021 IEEE symposium series on computational intelligence (SSCI). pp. 1–8. IEEE (2021)
8. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. ACM (2016)
9. Salvador, R.A., Saludares, M.I.: Autonomous driving via deep reinforcement learning. In: Proceedings of the Philippine Computing and Networking Conference (PCNaval). Quezon City, Metro Manila (2019)
10. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
11. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Gradcam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
13. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics: Results of the 11th international conference. pp. 621–635. Springer (2017)
14. Shi, C., Zhang, S., Lu, W., Song, R.: Statistical inference of the value function for reinforcement learning in infinite-horizon settings. Journal of the Royal Statistical Society Series B: Statistical Methodology **84**(3), 765–793 (2021)
15. Siboo, S., Bhattacharyya, A., Raj, R.N., Ashwin, S.H.: An empirical study of ddpg and ppo-based reinforcement learning algorithms for autonomous driving. IEEE Access **11**, 125094–125108 (2023)
16. Stable-Baselines3 Contributors: Reinforcement learning tips and tricks. https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html (2021), accessed: 2026-01-07
17. Stolz, R., Eichelbeck, M., Althoff, M.: Improving stochastic action-constrained reinforcement learning via truncated distributions. Tech. rep., Technical University of Munich (2023)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press (2018)
19. Wang, C., Aouf, N.: Explainable deep adversarial reinforcement learning approach for robust autonomous driving. IEEE Transactions on Intelligent Transportation Systems (2023)