Kolmogorov–Arnold Controllers: Toward Smooth and Safer Embedded Policies

Swarnava Dey
Artificial Intelligence, IIT Kharagpur and TCS Research, Tata Consultancy Services Ltd.
Kolkata, West Bengal, India, 700135
Email: swarnava.dey@tcs.com, ORCID: 0000-0002-3988-1445.

***Abstract*—Reinforcement learning (RL) controllers often rely on multilayer perceptrons (MLPs), whose sharp action responses can behave unpredictably when safety intervention is required. Kolmogorov–Arnold Networks (KANs), built from smooth spline operators, offer a more regular alternative. We compare MLP and KAN policies under two RL algorithms and two control tasks of differing difficulty, using identical training setups and a light supervisory mechanism that overrides actions near unsafe conditions. In the easier task, both architectures achieve comparable performance. In the harder, safety-critical setting, the MLP controller becomes unstable under supervision, while the KAN policy maintains consistent learning and achieves substantially lower unsafe-state rates. Our goal is not to claim superiority of either model, but to characterize their stability, safety profiles, and design trade-offs in the context of embedded AI controllers. Early results suggest that mid-sized KANs produce smoother activation patterns and safer trajectories while retaining sample efficiency comparable to equally parameterized MLPs.**

I. Introduction

Deep Reinforcement Learning (RL) policies [1] are increasingly deployed in industrial control [2], yet the dominant multilayer perceptron (MLP) architectures provide little inductive bias for smooth dynamics, exhibit numerical brittleness, and offer limited interpretability for safety-critical settings. Kolmogorov–Arnold Networks (KANs) replace affine transformations with learnable spline operators and have recently demonstrated enhanced functional stability and symbolic decomposability [3], [4], [5].

Neural-network controllers have long been studied as nonlinear feedback approximators [6], and RL has become a standard tool for synthesizing such controllers across continuous-control tasks [1]. However, safety during training and deployment remains a key weakness of standard RL, motivating the development of *shielding* mechanisms that override unsafe actions using temporal-logic or heuristic constraints [7], [8], [9], [10]. Shielding preserves modularity but implicitly assumes policies whose outputs behave smoothly and predictably—properties not naturally satisfied by MLPs.

This motivates examining whether KANs, with their smoother and more interpretable function classes, offer *practical* benefits for safety-aware control. Although KAN-based RL [11], [12], [13], [14] and KAN-based safe control [15], [16] have emerged independently, no prior work provides a systematic comparison of MLP and KAN controllers on accuracy–safety–efficiency trade-offs under an explicit shielding module.

Goal. We investigate whether KANs offer advantages over equally sized MLPs as controllers in safety-critical RL, focusing on (i) unsafe-state frequency, (ii) shield intervention rate, (iii) return stability, and (iv) embedded efficiency.

Experimental design. We compare matched-capacity MLP and KAN controllers under REINFORCE and PPO [1], [2], with identical actor–critic heads, identical training pipelines, and identical handcrafted safety shields. The study spans an easy benchmark (CartPole) and a harder one (`LunarLander-v3`), the latter requiring precise thrust modulation, impact damping, and stable approach dynamics.

Contributions.

1) A matched, fair comparison of KAN vs. MLP RL controllers under identical learning and safety conditions.
2) A mathematically defined shield for Lunar lander.
3) Evidence that KANs achieve smoother and safer control on hard tasks while retaining embedded efficiency.

II. Methodology

Tasks. We evaluate on *CartPole* (state $x = [x_p, \dot{x}, \theta, \dot{\theta}]$; continuous action u; safety $|\theta| \le \theta_{\max}$) and *LunarLander-v3* (state $s = (x, y, \dot{x}, \dot{y}, \theta, \dot{\theta}, l, r)$; 4 actions). All runs use identical settings, seeds, and matched feature-extractor capacity.

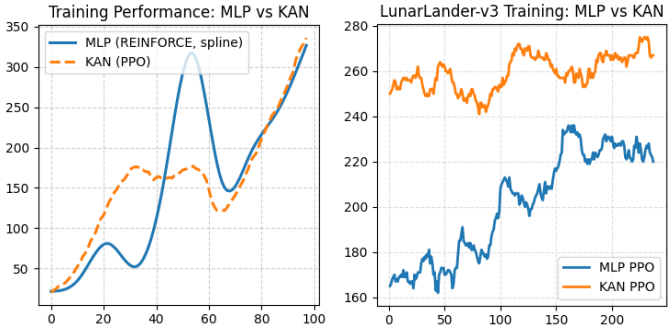
Architectures. The MLP uses layers $8 \to 24 \to 24 \to 32$. The KAN replaces these with two spline layers: KANLayer$(8, 64)$ and KANLayer$(64, 64)$ with grid=5, order $k = 3$. Actor and critic heads are identical: $\pi(a|s) = W_a z$, $V(s) = W_v z$, $z \in \mathbb{R}^{64}$. Total parameters and feature-dimension are matched (9472 parameters each, including PPO heads).

Shielding. For the harder task, we apply a lightweight supervisory override. Let a denote the action proposed by the policy, a' the executed action, and $s = (x, y, \dot{x}, \dot{y}, \theta, \dot{\theta})$ the system state. When s enters an unsafe region, defined as

$$\mathcal{U} = \{s : \ |\theta| > 0.3 \vee |\dot{y}| > 0.8 \vee (y < 0.05 \wedge (|\theta| > 0.15 \vee |\dot{x}| > 0.5))\}.$$

the supervisor replaces the action with a stabilizing thrust ($a' = 2$) with probability $p(s)$; otherwise $a' = a$. The probability: $p(s) = \alpha \, \mathrm{dist}(s, \mathcal{U})^{-1}$, increases smoothly as the state approaches the unsafe boundary, allowing intervention without freezing exploration.

Metrics. We report (i) mean return, (ii) return std., (iii) unsafe-state rate, (iv) shield-intervention rate, and (v) latency.



(a) CartPole (b) LunarLander-v3

Fig. 1: Training MLP vs. KAN under identical PPO pipelines. x-axis: normalized training progress, y-axis: mean rewards.

III. Results

Learning behaviour. Fig. 1 shows PPO learning curves. CartPole saturates quickly for all models, while LunarLander reveals clear architectural differences.

CartPole. In the easier task, REINFORCE exposes clear weaknesses of both architectures. The MLP achieves moderate returns (≈ 375) but does not improve with the shield, while the KAN attains higher peak returns (≈ 394) but exhibits large variance and becomes less stable when supervised. Under PPO, however, both models reliably reach the task limit (≈ 500) with zero unsafe states. Overall, for the simple task, algorithmic effects dominate, and architectural differences become negligible once a stable RL algorithm is used.

LunarLander (hard). Here architecture matters:
leftmargin=*

- **MLP (no shield):** mean 212.9, unsafe 0.157.
- **MLP (shield):** collapses (mean 5.0, huge variance, unsafe 0.120, interventions 0.119).
- **KAN (no shield):** mean 272.1, unsafe 0.045.
- **KAN (shield):** unsafe drops to 0.0038 (12× reduction), interventions only 0.030, with moderate return change.

Safety & stability. The shield destabilizes MLP but stabilizes KAN, indicating that KAN policies form smoother decision boundaries more compatible with a supervisory override. In contrast, MLPs respond sharply to small state changes, triggering oscillatory override patterns.

Efficiency. Both models have identical parameter count (9472). Latency is nearly identical: MLP 5.86ms vs. KAN 4.63ms. Thus, KAN safety improvements incur no embedded-system overhead.

IV. Conclusion

KANs provide smoother control, lower unsafe-state frequencies, and better compatibility with safety shields—especially in difficult tasks—while retaining identical size and latency to MLPs. These results suggest KANs are strong candidates for resource-constrained safe RL. Future work includes shield synthesis tuned to KAN smoothness and automated KAN architecture design.

References

[1] B. Modi and H. B. Jethva, "Reinforcement learning with neural networks: A survey," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*, S. C. Satapathy and S. Das, Eds. Cham: Springer International Publishing, 2016, pp. 467–475.

[2] P. Unnikrishnan and P. Vijayakumar, "Intelligent control using neural networks & reinforcement learning," *International Journal of Applied Engineering Research*, vol. 10, pp. 1817–1831, 01 2015.

[3] Y. Peng, Y. Wang, F. Hu *et al.*, "Predictive modeling of flexible ehd pumps using kolmogorov–arnold networks," *Biomimetic Intelligence and Robotics*, vol. 4, no. 4, p. 100184, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667379724000421

[4] S. Wang, J. Chen, X. Liu, T. Zhang, X. Chai, Q. Lu, D. Shen, and H. He, "Kriging to kolmogorov-arnold network model accelerated discovery of oxygen control strategy in lead-based fast reactors," *Nature Communications*, vol. 16, no. 1, p. 9774, Nov 2025. [Online]. Available: https://doi.org/10.1038/s41467-025-64747-7

[5] Z. Cheng, T. Yu, G. Jia, and Z. Shi, "A physics-informed neural network-based method for dispersion calculations," *International Journal of Mechanical Sciences*, vol. 291-292, p. 110111, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020740325001973

[6] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systems: a survey," *Automatica*, vol. 28, no. 6, p. 1083–1112, Nov. 1992. [Online]. Available: https://doi.org/10.1016/0005-1098(92)90053-I

[7] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[8] H. Odriozola-Olalde, M. Zamalloa, N. Arana-Arexolaleiba, and J. Perez-Cerrolaza, "Towards robust shielded reinforcement learning through adaptive constraints and exploration: The fear field framework," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110055, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197625000557

[9] B. Könighofer, R. Bloem, N. Jansen, S. Junges, and S. Pranger, "Shields for safe reinforcement learning," *Commun. ACM*, vol. 68, no. 11, p. 80–90, Oct. 2025. [Online]. Available: https://doi.org/10.1145/3715958

[10] E. H.-D. l. Court, F. Belardinelli, and A. W. Goodall, "Probabilistic shielding for safe reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 15, pp. 16 091–16 099, Apr. 2025. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/33767

[11] S. Somvanshi, S. A. Javed, M. M. Islam, D. Pandit, and S. Das, "A survey on kolmogorov-arnold network," *ACM Comput. Surv.*, vol. 58, no. 2, Sep. 2025. [Online]. Available: https://doi.org/10.1145/3743128

[12] V. A. Kich, J. A. Bottega, R. Steinmetz, R. B. Grando, A. Yorozu, and A. Ohya, "Kolmogorov-arnold networks for online reinforcement learning," in *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, 2024, pp. 958–963.

[13] J. Wu, R. Du, and Z. Wang, "Deep reinforcement learning with dual-q and kolmogorov–arnold networks for computation offloading in industrial iot," *Computer Networks*, vol. 257, p. 110987, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624008193

[14] J. Huang, R. Zhou, M. Li, H. Li, Y. Liu, and X. Song, "From black-box to white-box: Interpretable deep reinforcement learning with kolmogorov-arnold networks for autonomous driving," *Transportation Research Part C: Emerging Technologies*, vol. 182, p. 105386, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X25003900

[15] X. Zhang, N. Lv, W. Lin, and Z. Ding, "Formal synthesis of safe kolmogorov-arnold network controllers with barrier certificates," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, ser. IJCAI '25, 2025. [Online]. Available: https://doi.org/10.24963/ijcai.2025/35

[16] X. Zhang, J. Xu, Y. Wang, D. Xiang, W. Lin, and Z. Ding, "Formal synthesis of barrier certificates using fourier kolmogorov-arnold network," in *AAAI'25/IAAI'25/EAAI'25*. AAAI Press, 2025. [Online]. Available: https://doi.org/10.1609/aaai.v39i1.32101