Counterfactual Explanations for Reinforcement Learning Agents using Kolmogorov-Arnold Networks

Agustín Valencia*‡, Michele Antonio D'Alto† and Ahmad Terra*†
{agustin.valencia, ahmad.terra}@ericsson.com
‡Corresponding author
*Ericsson Research, Stockholm, Sweden
†KTH, Stockholm, Sweden

***Abstract*—We propose an explainable reinforcement learning framework for radio access network control that implements Proximal Policy Optimisation (PPO) using Kolmogorov-Arnold Networks (KANs), enforcing sparsity, pruning, and symbolification so the learned actor exposes interpretable closed-form logits. Moreover, we provide a method to answer questions like "what minimal input changes, would alter the model's output?" by exploiting the symbolic form of the policy and thus removing the need of any post-hoc approximations. For such, we construct compact sets of counterfactual explanations that flip the selected action while remaining feasible and near the query state, balancing validity, proximity, and diversity. We validate our method in a sectorised remote electrical tilt simulation, where KAN-based agents match or slightly exceed multi-layer perceptron (MLP) baselines. Furthermore, we show that after symbolification, the policy does not suffer performance degradation and becomes auditable through short analytic expressions. As the procedure depends only on a feasibility map and feature scales, its applicability extends in a wide range of other control tasks.**

***Index Terms*—RL, XRL, counterfactual explanations, Kolmogorov–Arnold Networks, PPO, radio access networks, remote electrical tilt, Cholesky decomposition**

I. Introduction

The transition towards 6G will require autonomous, adaptive, and energy-efficient Radio Access Networks (RANs), where Artificial Intelligence (AI) is expected to play a central role in continuous optimisation of coverage, capacity, and quality of service [1]. Among learning paradigms, Reinforcement Learning (RL) is especially suitable for network control because it can optimise long-horizon objectives directly from interaction, as evidenced by applications such as beamforming [2] and antenna tilt optimisation [3] by exploiting the exceptional, but black-boxed, function approximation capabilities of classical deep Artificial Neural Networks (ANNs) layers –Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers–. Yet, deploying RL in live networks remains challenging: decisions must respect strict safety and operational constraints, and stakeholders require transparent behaviour to satisfy internal governance and emerging regulation as the EU AI Act [4]. Under this regulatory framework, particularly for high-risk applications, AI systems must be transparent and provide explanations that allow human oversight to understand the reasoning behind specific outputs. Counterfactuals (CFs) specifically mitigate these risks by providing actionable "what-if" scenarios by demonstrating which minimal changes in the network state would lead to a different control decision, operators can verify if the policy aligns with safety guidelines or identify potential biases in the decision-making logic.

Explainability is thus not a peripheral matter but a deployability concern. Post-hoc explanations, though useful as diagnostics, struggle to provide guarantees in non-stationary and highly coupled wireless environments. A complementary path is intrinsic interpretability: designing policies whose structure can be inspected, audited, and reasoned about before, during and after operation. KANs replace fixed activations with learnable spline operators, enabling sparse representations and symbolic extraction; our goal is to produce actors whose logits admit compact, closed-form expressions that support policy auditing, compliance review and knowledge transfer. In this study, we investigate KANs as actor and critic within Proximal Policy Optimisation (PPO).

Operational teams, however, also need actionable guidance around a policy's decisions. We therefore complement intrinsic interpretability with CFs explanations tailored to network control, i.e., finding local alternative states that remain feasible and close to the current operating point, yet would flip the policy's chosen action, so it is possible to discern other plausible courses of actions. Moreover, we also encourage CFs to be as diverse as possible to cover a wide range of possibilities. In the state of the art, such operation is computationally expensive, and therefore in this work we adopt a log- determinant diversity objective together with an incremental update based on rank-1 Cholesky factors, which improves the computational efficiency of the published algorithms.

We study these ideas in the context of a control task

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors thank Franco Ruggeri, Ioanna Mitsioni, and Maxime Bouton for providing valuable feedback that helped improve the clarity of the paper.

for Remote Electrical Tilt (RET), a representative task in RANs featuring coupled interference, evolving traffic and hard feasibility constraints.

The main contributions of this paper are: (i) being, at the best of authors' knowledge, the first KAN application study for RAN, (ii) achieving intrinsically interpretable controllers that compete in performance with the black-box state-of-the-art controllers in a RET task, (iii) leveraging KAN-based symbolification to guarantee the policy is well-defined in all its domain with no discontinuities, (iv) showing that the symbolification step of a KAN-policy from its spline-based form does not affect the agent's stability nor its control performance in the context of RET control, and (v) generating valid, proximal and diverse CFs in an optimal manner without requiring post-hoc approximations.

II. Background

A. Reinforcement Learning and PPO

In RL, an agent interacts with an environment over discrete time steps, receiving observations s_t, taking actions a_t according to a policy $\pi_\theta(a_t \mid s_t)$ parameterised by θ, and receiving rewards r_t. The objective is to maximise the expected discounted return

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{1}$$

Policy-gradient methods directly optimise $J(\theta)$ via stochastic estimates of $\nabla_\theta J(\theta)$, often with variance reduction through baselines and advantage estimators [5], [6]. PPO stabilises policy-gradient updates by introducing a surrogate objective. Let

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \tag{2}$$

denote the probability ratio between the updated and old policies. The surrogate objective is based on $r_t(\theta)\hat{A}_t$, where $\hat{A}_t$ is an estimate of the advantage function [7], but is clipped to constrain $r_t(\theta)$ within a trust region. This clipping prevents destructive updates while still allowing multiple epochs of minibatch optimisation [8], [9]. In practice, PPO is commonly implemented in an actor–critic form with a value-function baseline to reduce gradient variance, and $\hat{A}_t$ is implemented using Generalized Advantage Estimation (GAE) [10] which exposes $\lambda \in [0, 1]$ to control the variance-bias trade-off, where $\lambda = 0$ implies low variance and high bias –pure TD(0)– and $\lambda = 1$ implies high variance and low bias –pure Monte Carlo–.

B. Diverse Counterfactual Explanations in RL

CF explanations answer questions of the type: "what minimal, feasible changes to an input x would alter a model's output y?". Diverse Counterfactual Explanations (DiCE) [11], [12] generates not one but multiple *diverse* CFs by optimising for validity (decision flips), proximity (small changes), and diversity (coverage of distinct alternatives). Recent extensions improve the robustness of generated CFs to small perturbations and refine the multi-objective trade-offs [13]. Although primarily developed for supervised models, the same principles can support RL analysis by proposing alternative actions to reach different states while remaining close in state–action space; this enables more transparent behaviour audits and policy debugging. Furthermore, in the present work we propose an improvement in the algorithm itself, i.e. independent of our work on KAN, to optimally find the CFs faster.

C. Kolmogorov–Arnold Networks

The architecture of KANs is motivated by the Kolmogorov-Arnold Representation Theorem (KART) [14], [15], which proves that any continuous multivariate function on $[0, 1]^n$ can be decomposed into finite compositions of continuous univariate functions and addition (see Appendix A for the formal statement). Unlike the universal approximator theorem, the KART implies equivalency, not approximation.

KANs reinterpret this decomposition into a learnable neural architecture: each connection is endowed with a univariate learnable function –instead of a scalar weight as in MLPs– and nodes sum the outputs of those functions without further non-linearities [16]. The result is a model where non-linearity and coupling are captured by these per-edge functions rather than by fixed activation units.

1) KAN Sparsity and Pruning: KANs enforce sparsity through a composite regularisation loss (detailed in Appendix A). Each edge function φ is a B-spline whose L_1 activation magnitude, entropy, coefficient magnitude, and smoothness are penalised. The total training loss combines the task objective with these penalties:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \sum_{l=0}^{L-1} \big(&\mu_1 \|\Phi_l\|_1 + \mu_2 \mathcal{H}(\Phi_l) \\ &+ \mu_3 \mathcal{C}(\Phi_l) + \mu_4 \mathcal{C}_{\text{diff}}(\Phi_l)\big)\end{aligned} \tag{3}$$

where λ balances task performance versus sparsity. After training, edges with average activation below a threshold τ are pruned, yielding a sparse graph that aggregates only meaningful inputs.

2) KAN Symbolification: For each remaining active spline φ, we search for a symbolic representation by fitting an affine candidate f (linear, polynomial, trigonometric, exponential, logarithmic, etc.) to pre/post-activation pairs (x, y) via

$$y \;\approx\; c\, f(ax + b) + d, \tag{4}$$

selecting $(a, b, c, d) \in \mathbb{R}^4$ and accepting the match when R^2 exceeds a chosen threshold. Replacing splines with symbolic forms produces a transparent model whose components can be plotted and analysed.

3) KANs applications: Empirically, Liu et al. show that small KANs can match or surpass larger MLPs in regression tasks and PDE solving, while offering interpretability through visualization of the learned edge functions [16]. Subsequent works have begun exploring KANs in reinforcement learning contexts: for example, a KAN-based function approximator was used in PPO in continuous control tasks [17], and even in

load balancing control tasks with interpretable policy extraction [18]. Other variants such as PRKAN (parameter-reduced KANs) [19] and Wav-KAN (wavelet-based KANs) [20] are being developed to enhance parameter efficiency or expressive flexibility.

Differentiable Decision Trees (DDTs) [21] offer an alternative interpretable baseline by parameterising soft splits with sigmoid functions, enabling end-to-end gradient-based training within the RL loop—unlike post-hoc distillation methods [22] that introduce a performance gap. We include DDTs alongside KANs and MLPs (Sec. III-B).

III. Method

A. Agents Architecture

We adopt an actor–critic PPO backbone for both the KAN-based agent and the MLP baseline. KAN layers are implemented with PyKan [16] (splines, gradients, optional symbolification) while we use a custom implementation of the PPO loop for better control over rollout collection, GAE, clipping, and optimisation. Both agents share rollout length, minibatching, advantage normalisation, value-loss weighting, entropy bonus, early stopping and random seeds.

1) KAN-Based Agent: Following [17], actor and critic have stacked KAN layers. We train with sparsity-aware regularisation, prune edges with low activity, and then symbolify surviving univariate splines (see II-C1, II-C2). The resulting sparse, symbolic networks are used for analysis. In our method we split the KAN training in a spline-training phase, where the splines are fitted, and a symbolic training phase, where we keep fine-tuning the KAN after the symbolic policy has been obtained.

2) MLP Baseline: The baseline is sized for fair comparison against KAN agents. We use a two-hidden-layer MLPs with ReLU activations, and use identical PPO settings.

B. Agents Training

Both agents are trained with PPO using

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big] - \beta\,\mathcal{H}(\pi_\theta) + c_v\,\text{MSE}(V_\theta(s_t), R_t) \tag{5}$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$, $\hat{A}_t$ is GAE, $\mathcal{H}$ is policy entropy, $R_t = \hat{A}_t + V_\theta(s_t)$ and V_θ the critic value function.

For KANs we add

$$\mathcal{L}_{\text{KAN}} = \mathcal{L}_{\text{PPO}} + \lambda_{\text{KAN}}\mathcal{R}_{\text{KAN}} \tag{6}$$

with

$$\mathcal{R}_{\text{KAN}} = \sum_l \Big\{\mu_1\|\Phi_l\|_1 + \mu_2\mathcal{H}(\Phi_l) + \mu_3\mathcal{C}(\Phi_l) + \mu_4\mathcal{C}_{\text{diff}}(\Phi_l)\Big\} \tag{7}$$

Pruning uses threshold τ; symbolification following II-C2.

C. Counterfactuals

We consider a trained a multi-class model for C classes that maps inputs $x \in \mathbb{R}^d$ to logits $a(x) \in \mathbb{R}^C$ and predicts

$$y = g(x) = \underset{i\in\{1,\dots,C\}}{\arg\max}\ a_i(x) \tag{8}$$

Given a query instance x_0, a counterfactual c is a feasible state that remains close to x_0, this is, $c \in [x_0 \pm \varepsilon], \{x_0, \varepsilon\} \in \mathbb{R}^d$ yet changes the model's prediction. Formally [23]:

$$\begin{aligned} y_0 &:= g(x_0) \\ y^*(c) &:= g(c) \\ y^*(c) &\neq y_0 \end{aligned} \tag{9}$$

Following the DiCE works [11], we return a set of k counterfactuals $C = \{c_1, \dots, c_k\}$ that jointly trade off validity, proximity, and diversity.

1) Justification in RL Context: While counterfactuals are traditionally applied to i.i.d. supervised learning, their use in RL is justified by our objective of explaining the *agent's policy* rather than the environment's transition dynamics. In safety-critical domains like RAN control, human operators are primarily concerned with the immediate reasoning behind a specific action (e.g., "Why did the agent downtilt in this cell?"). By treating the policy π as a mapping $g : \mathcal{S} \to \mathcal{A}$, we can leverage CF methods to identify the minimal state perturbations that would result in a different control decision. This provides a local explanation of the policy's decision boundary, which is complementary to global interpretability provided by the symbolic form of KANs. To ensure that counterfactual states remain physically realisable, we restrict candidates to historically observed states (Sec. III-C2).

2) Feasibility: The feasibility of a CF candidate is characterised by the set of features that can be perturbed. Thus, let us encode the samples through a partition of the feature indices into two sets. The set M contains the mutable features, whereas the set F contains the fixed coordinates that must be preserved due to domain or operational constraints. We write $x = \{x_M, x_F\}$ accordingly. In RL, maintaining feasibility is paramount to avoid generating infeasible states. We restrict the search to a candidate pool drawn from a reference set $\mathcal{D} \subset \mathbb{R}^d$, which in principle could be constructed via descriptive or generative approaches. In this study we simply set $\mathcal{D}$ equal to the set of states observed during training. The feasible pool for a query x_0 is then

$$\begin{aligned} S_{x_0} &= \big\{\, x \in \mathcal{D} :\ x_F = x_{0,F}\ \wedge\ \text{constraints}\,\big\}, \\ n &:= |S_{x_0}|. \end{aligned} \tag{10}$$

3) Validity Scoring: For a candidate $c \in S_{x_0}$, validity is quantified through a multi-class margin defined as

$$\Delta(c) = \max_{j\neq y_0} a_j(c) - a_{y_0}(c), \tag{11}$$

$$\ell_{\text{val}}(c) = (\delta - \Delta(c))_+ \tag{12}$$

where $\delta > 0$ defines our confidence margin. The loss $\ell_{\text{val}}(c)$ equals zero precisely when $y^*(c) \neq y_0$ and the margin is at least δ. See Appendix B for proof.

4) Proximity Scoring: The proximity between $u, v \in \mathbb{R}^d$ is evaluated exclusively over the subset of mutable features $M \subseteq \{1, \ldots, d\}$. It is defined as

$$d_M(u, v) = \frac{1}{|M|} \sum_{j \in M} \frac{|u_j - v_j|}{w_j} \tag{13}$$

where each $w_j > 0$ indicates a per-feature normalisation coefficient. To account for heterogeneity across feature magnitudes, we set w_j as the standard deviation of feature j estimated from $\mathcal{D}$.

5) Diversity Scoring: To encourage non-redundant CFs sets, diversity is modelled through a kernelised log-determinant criterion. For a set $C = \{c_1, \ldots, c_k\}$, we define

$$K_{ij}(C) = \frac{1}{1 + d_M(c_i, c_j)} \tag{14}$$

$$\mathrm{div}(C) = \log\det K(C) \tag{15}$$

where $K(C)$ denotes the similarity matrix constructed from the distance $d_M(\cdot, \cdot)$ defined in (13). Maximising $\log\det K(C)$ thus, favours sets whose members are well distributed and informationally complementary.

In summary, with weighting coefficients $\lambda_1, \lambda_2 > 0$, the overall objective maximises $G(C) = -F(C)$, where

$$F(C) = \frac{1}{k} \sum_{c \in C} \ell_{\mathrm{val}}(c) \tag{16}$$

$$+ \lambda_1 \frac{1}{k} \sum_{c \in C} d_M(c, x_0) \tag{17}$$

$$- \lambda_2 \, \mathrm{div}(C) \tag{18}$$

$$\text{s.t. } |C| = k \quad , \quad C \subseteq S_{x_0} \tag{19}$$

The term in (16) enforces validity, the term in (17) promotes proximity to the query instance x_0, and (18) encourages diversity among the counterfactuals within C.

6) Interpretability Metric: To quantify the interpretability of the extracted symbolic policies, we adopt the *Effective Complexity* (C_{eff}) defined as the total number of nodes in the expression tree of the symbolic form. Each leaf (input variable or numerical coefficient) and each operator ($+$, $\times$, $(\cdot)_+$, $|\cdot|$) counts as one node. This metric, standard in the symbolic regression literature [24], serves as a proxy for the cognitive load required for a human expert to audit the control law: lower C_{eff} implies a more compact, readable expression. Because black-box models such as MLPs have no closed-form expression, C_{eff} is only defined for architectures that produce symbolic policies, making it a discriminating measure of interpretability.

7) Selection and Fast Diversity via Cholesky: Following [11], we greedily add to C the candidate with the largest marginal gain $G(C \cup \{i\}) - G(C)$. Let $K(C) = LL^T$ be the Cholesky factorisation of the kernel matrix. For a new candidate i with similarity vector $d = (K_{ij})_{j \in C}$, we prove (Appendix C) that:

$$\det K(C \cup \{i\}) = \det K(C) \left(1 - d^T K(C)^{-1} d\right) \tag{20}$$

By solving $Lu = d$ we obtain $d^T K(C)^{-1} d = u^T u$ (Appendix D). Setting $\gamma := 1 - u^T u$, the log-determinant increment becomes $\log\det K(C \cup \{i\}) = \log\det K(C) + \log\gamma$, and the Cholesky factor updates via (Appendix E):

$$L_{\mathrm{new}} = \begin{bmatrix} L & 0 \\ u^T & \sqrt{\gamma} \end{bmatrix}, \quad L_{\mathrm{new}} L_{\mathrm{new}}^T = K(C \cup \{i\}) \tag{21}$$

This reduces the diversity update from $\mathcal{O}(|C|^3)$ to $\mathcal{O}(|C|^2)$ per iteration with linear memory. The full procedure is given in Algorithm 1 (Appendix I).

IV. Experimental Configuration

To assess the method we conduct a set of simulations that emulate a RAN with RET control. The simulator captures propagation, user spatial distributions, sectorisation, and inter-sector interference in a controlled and reproducible environment. Our focus is the optimisation of electrical downtilt to optimise coverage and quality while maintaining service stability.

Across cellular deployments, tilt controls the main-lobe direction of a sector antenna and thereby governs both coverage and interference. A typical Base Station (BS) comprises three sector antennas that jointly span approximately 120° each. Adjusting their tilts modulates the experienced signal conditions for nearby User Equipments (UEs). Figure 1 provides a schematic view.

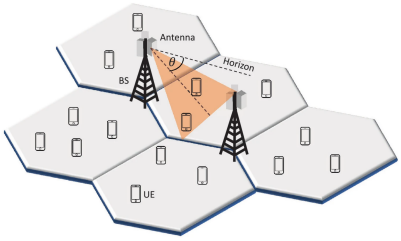


Fig. 1. Schematic illustration of a three-sector base station with electrical downtilt from [25].

1) State and Action Space: Each sector behaves as an agent observing three features for U users: average Reference Signal Received Power (RSRP) $\overline{\mathrm{RSRP}}(\theta) = \frac{1}{U} \sum_i \mathrm{RSRP}_i(\theta)$, average Signal to Interference plus Noise Ratio (SINR) $\overline{\mathrm{SINR}}(\theta) = \frac{1}{U} \sum_i \mathrm{SINR}_i(\theta)$, and current electrical downtilt $\theta \in \{0, \ldots, 15\}$ degrees. For numerical stability, measurements are standardised:

$$\widetilde{z} = \frac{z - \mu_z}{\sigma_z}, \qquad z \in \{\overline{\mathrm{RSRP}}, \overline{\mathrm{SINR}}\}, \tag{22}$$

and the tilt is mapped to $[-1, 1]$:

$$\widetilde{\theta} = 2 \, \frac{\theta - \theta_{\min}}{\theta_{\max} - \theta_{\min}} - 1, \quad \theta_{\min} = 0, \ \theta_{\max} = 15 \tag{23}$$

The action set comprises incremental tilt adjustments $a \in \mathcal{A} = \{-1, 0, +1\}$ (one-degree decrease, hold, or increase).

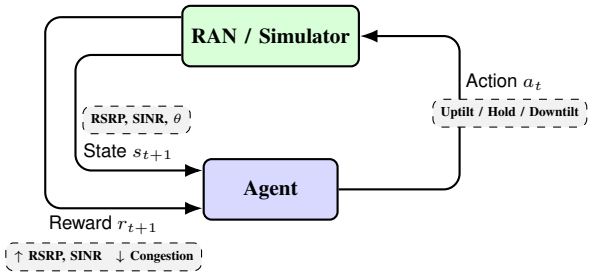


Fig. 2. RL loop: the agent observes the network state, performs tilt actions and receives updates based on quality metrics and traffic congestion

2) Reward: The per-step reward is the fraction of users with adequate coverage and quality:

$$r = \frac{1}{U}\sum_{i=1}^{U} \mathbf{1}\{\text{RSRP}_i \geq \tau_{\text{RSRP}} \ \wedge \ \text{SINR}_i \geq \tau_{\text{SINR}}\} \qquad (24)$$

This encourages improved reach while implicitly penalising over-aggressive tilts via interference. The environment is modelled as an Markov Decision Process (MDP) with $\mathcal{S} = \{[\overline{\text{RSRP}}, \overline{\text{SINR}}, \theta]\}$, $\mathcal{A} = \{-1, 0, +1\}$, $\mathcal{R} \subseteq [0, 1]$. Fig. 2 summarises the control loop.

The simulated layout (Appendix J) comprises multiple BSs with heterogeneous indoor/outdoor UE distributions. The task is challenging due to the coupled interference landscape: initial tilts are randomised per episode, and agents must balance coverage extension against inter-sector interference.

3) Relevance to the Method: This simulated RAN serves as a stress test for the methodology. The symbolic KAN actor trained with PPO provides closed-form logits $a_i(x)$ for post-hoc analysis. Counterfactual sets are then generated as in Sec. III-C. The Cholesky-based diversity updates enable fast selection over large candidate pools produced by the simulator, preserving both quality and diversity while keeping the computing-time practical.

A. Hyperparameters and Training Setup

To assess generalizability, we additionally evaluate on CartPole-v1 (Appendix K). We evaluate on a simulated RAN with RET control having episodes of length 20. Both MLPs and KANs use identical PPO hyperparameters: γ=0.99, λ=0.95, ϵ=0.2, c_v=0.5, β=0.01, learning rate 10^{-3}, 2000 steps/rollout, 6 PPO epochs and minibatches of 64. KAN models train for 5000 episodes with pruning every 1000 episodes at τ=0.01, then the top configuration is fine-tuned for 1000 episodes with symbolification. Each final setting is repeated over multiple random seeds to ensure statistical robustness.

V. Results and Discussion

A. Selected Architectures

For the MLP baselines we test hidden widths $[64, 64]$ (MLP-64), $[128, 128]$ (MLP-128), and $[256, 256]$ (MLP-256) for both actor and critic. As interpretable baselines, we include [21] with depths 3 (DDT-3) and 4 (DDT-4), which provide extractable decision rules via soft-to-hard discretisation. KAN sweeps cover grid size $\{5, 8\}$, spline degree $\{3, 5\}$, hidden neurons $\{0, 3, 5, 8\}$, and regularisation weights $(\mu_1, \mu_2) \in \{0, 1, 2\}$, $(\mu_3, \mu_4) \in \{0, 0.2, 0.5\}$. We retain two configurations: **KAN-0** ($[3, 3]$ topology, 9 edges) for maximal compactness, and **KAN-8** ($[3, 8, 3]$ topology, up to 33 edges before pruning) for greater expressivity. Both use grid 5, degree 3; the best actor regularisation is $(\mu_1, \mu_2, \mu_3, \mu_4)$=$(2, 1, 0.2, 0.2)$.

A detailed comparison of parameter counts and interpretability characteristics is deferred to Table III in Sec. V-D; notably, the KAN agent is significantly more compact than both the MLP and DDT baselines.

B. Learning Performance

Fig. 3 shows $\mu \pm 1\sigma$ of the cumulative reward per training episode for all evaluated architectures. KAN-8 converges to the highest reward ($\approx$17.5), matching or slightly exceeding MLP-64, MLP-128, and DDT-4. The simpler KAN-0 (no hidden layer) plateaus at $\approx$16.5, comparable to DDT-3 ($\approx$16.3) and MLP-256 ($\approx$16), but with the advantage of closed-form symbolic extraction. Fig. 4 shows per-episode training times: most architectures fall within 8.5–9.6 s, as the simulation dominates wall-clock cost. DDT-4 is the slowest ($\approx$10.9 s) due to its deeper tree. KAN-8 adds only $\approx$9% overhead relative to MLP-64, demonstrating that KAN interpretability imposes negligible computational cost in simulation-heavy RL settings.

Table I reports converged RAN performance metrics (last 20% of training episodes). We define convergence as the first episode at which a 50-episode rolling average of the cumulative reward reaches 95% of its final value. All seven architectures achieve comparable coverage (0.81–0.86) and mean SINR ($\approx$11–13 dB). KAN-8 converges fastest (262 episodes), followed by MLP-256 (253). KAN-0, despite its simpler structure, achieves the highest hold fraction (0.40), suggesting more conservative tilt control. DDT-4 and MLP-128 attain competitive coverage ($\approx$0.85) but converge more slowly (552 and 368 episodes, respectively). Overall, the domain-level behaviour is similar across architectures, confirming that KAN's interpretability benefit does not come at the cost of operational performance.

Let us recall that our proposed learning algorithm for the KAN-based agents establishes a spline-based learning phase and then a symbolic fine-tuning phase. Fig. 5 compares evaluation rewards before and after symbolification for both KAN architectures (100 episodes per seed, pooled across seeds). KAN-8 retains or slightly improves its reward distribution after symbolification (μ=16.5 vs. 14.3), while KAN-0 remains stable ($\mu\approx$11). These results confirm that the symbolic policy preserves the performance characteristics of the original spline-based network.

C. Symbolic Policy

Eq. (25) shows a representative logit of the KAN-0 symbolic policy (full formula in Appendix L):

$$a_{-1^\circ} = 0.34\sin(0.96\,\widetilde{s}+5.29) - 1.57\sin(0.52\,\widetilde{\theta}-3.88) + 0.68 \qquad (25)$$

TABLE I
CONVERGED RAN METRICS (LAST 20% OF TRAINING, $\mu \pm \sigma$ ACROSS SEEDS).

KAN Architectures		
Metric	**KAN-8**	**KAN-0**
Mean SINR (dB)	12.0 ± 0.6	11.9 ± 0.6
Coverage	0.843 ± 0.014	0.827 ± 0.009
Tilt changes / ep	14.0 ± 2.1	$\mathbf{12.0 \pm 1.7}$
Hold fraction	0.30 ± 0.10	$\mathbf{0.40 \pm 0.09}$
Convergence (ep)	262 ± 24	359 ± 98
MLP Baselines		
Metric	**MLP-64**	**MLP-128**
Mean SINR (dB)	$\mathbf{12.9 \pm 1.1}$	11.5 ± 0.0
Coverage	$\mathbf{0.858 \pm 0.007}$	0.834 ± 0.000
Tilt changes / ep	12.9 ± 1.1	15.1 ± 1.1
Hold fraction	0.35 ± 0.05	0.25 ± 0.05
Convergence (ep)	471 ± 101	368 ± 100
Metric	**MLP-256**	
Mean SINR (dB)	10.9 ± 0.8	
Coverage	0.832 ± 0.033	
Tilt changes / ep	12.8 ± 1.4	
Hold fraction	0.36 ± 0.07	
Convergence (ep)	$\mathbf{253 \pm 35}$	
DDT Baselines		
Metric	**DDT-3**	**DDT-4**
Mean SINR (dB)	11.2 ± 0.3	12.5 ± 0.0
Coverage	0.810 ± 0.014	0.855 ± 0.004
Tilt changes / ep	14.8 ± 1.0	15.2 ± 0.5
Hold fraction	0.26 ± 0.05	0.24 ± 0.03
Convergence (ep)	322 ± 154	552 ± 58

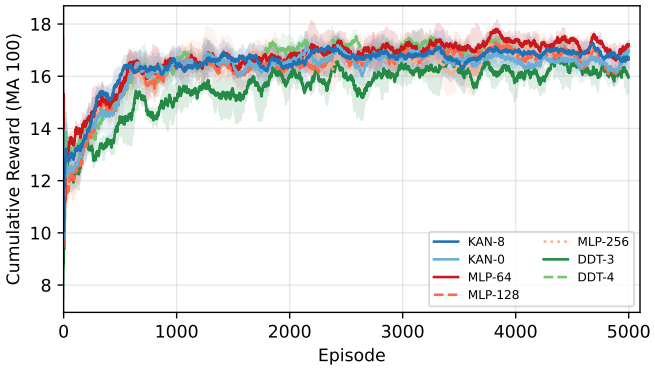


Fig. 3. Cumulative reward per episode ($\mu \pm 1\sigma$, 100-episode smoothing). KAN-8, MLP-64/128, and DDT-4 converge to ≈17; KAN-0, DDT-3, and MLP-256 plateau at ≈16.

where $\widetilde{s}$, $\widetilde{\theta}$ denote normalised SINR and tilt (Eqs. (22)–(23)). Analogous expressions for a_{+0° and a_{+1° involve sin, cos, and x^2 nonlinearities over all three features; the policy selects $a^* = \arg\max\{a_{-1^\circ}, a_{+0^\circ}, a_{+1^\circ}\}$.

This closed-form expression enables analytical verification (no division by zero or undefined operations) and direct inspection of decision boundaries. Fig. 6 visualises the KAN-8 spline-based policy regions over the $(\mathrm{SINR}, \mathrm{RSRP})$ plane for different tilts θ (red: downtilt, green: hold, blue: uptilt), revealing topological structure—such as the progressive emergence of downtilt at high θ—that would remain hidden in black-box representations.

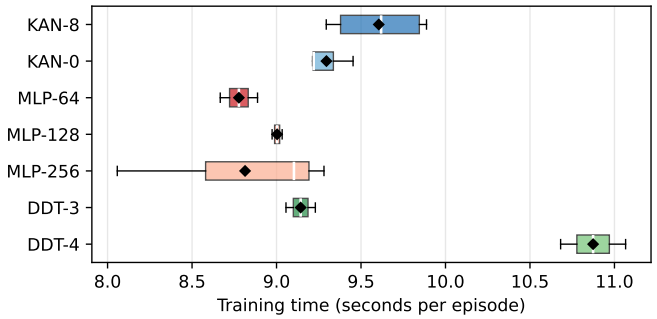


Fig. 4. Per-episode training time (seconds) across architectures. The simulation dominates wall-clock cost; most architectures fall within 8.5–9.6 s/episode, with DDT-4 at ≈10.9 s.

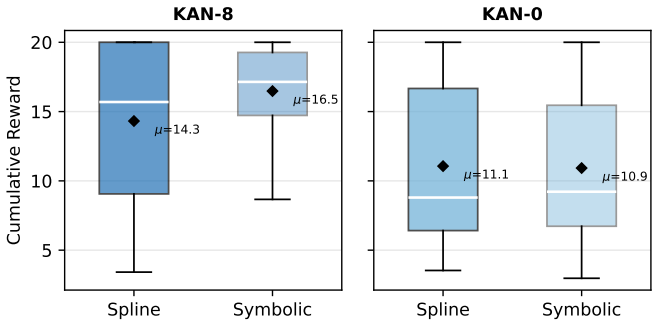


Fig. 5. Evaluation reward before (spline) vs. after (symbolic) symbolification for KAN-8 and KAN-0 ($\mu \pm \sigma$ across seeds, 100 eval episodes each).

D. Discussion and Practical Guidance

1) Symbolification Fidelity and Limitations: While symbolification (Eq. (25)) provides human-interpretable control laws, it is an approximation of the underlying spline-based network. Mismatches can occur if the library of symbolic functions (e.g., linear, sine, exp) cannot perfectly capture a complex spline shape.

Table II quantifies the fidelity of the symbolic approximation for both KAN variants. The no-hidden-layer variant (KAN-0) achieves substantially better fidelity than the deeper KAN-8: its *Action Agreement Rate* (AAR)—the fraction of an evaluation grid where spline-based and symbolic actors select the same action—averages 88.2% vs. 65.1%, and the mean logit RMSE is 0.99 vs. 36.5. The additive structure of KAN-0 (9 univariate edges, no hidden layer) makes each activation function a clean target for symbolic fitting. In contrast, KAN-8's two-layer composition amplifies small per-edge errors through 12–42 active edges, shifting decision boundaries more substantially.

Regarding *cross-seed stability*, all seeds produce distinct symbolic formulas for both architectures: the stochastic nature of both training and the symbolic regression library search means that different seeds converge to structurally different expressions. This is consistent with prior observations that KAN symbolic forms are sensitive to initialisation, and suggests that the specific formula in Eq. (25) should be interpreted as one

TABLE II
SYMBOLIFICATION FIDELITY ($\mu \pm \sigma$, EVALUATED ON A 100×100 SINR–RSRP GRID AT 6 TILT VALUES).

Metric	KAN-0	KAN-8
Active edges	9	34.5 ± 15.0
Action Agreement Rate	$88.2 \pm 7.2\%$	$65.1 \pm 16.1\%$
Logit RMSE	0.99 ± 0.37	36.5 ± 32.4

representative instance rather than a unique solution.

2) Effective Complexity and Interpretability: Table III compares the parameter counts and interpretability characteristics across all architectures. KAN-8's actor parameter count (888 ± 56) varies across seeds because pruning removes different edges depending on each run's training trajectory; the two-layer topology ($3\to8\to3$, up to 48 edges) provides redundancy that pruning exploits differently. In contrast, KAN-0's single-layer topology ($3\to3$, exactly 9 edges) leaves no edges to prune, so its parameter count is deterministic. For KAN-0, the expression tree contains $C_{\text{eff}}=59$ nodes across the three action logits (Appendix L). KAN-8's deeper topology yields $C_{\text{eff}}=323\pm165$, reflecting the larger number of composed symbolic edges (12–42 after pruning). DDTs offer an alternative form of interpretability: their soft splits can be discretised into hard decision trees with 7 (DDT-3) or 15 (DDT-4) internal nodes, yielding human-readable if–then rules. However, unlike KAN's closed-form expressions, DDT rules do not support analytical operations such as computing $\partial a/\partial\,\text{SINR}$. In contrast, MLPs are black-box mappings with 4k–68k actor parameters and no closed-form expression, making C_{eff} undefined. Beyond compactness, the KAN symbolic form enables *analytical verification*: a domain expert can directly compute partial derivatives to confirm that the policy responds monotonically to signal quality, or inspect the decision boundaries for pathological regions—capabilities that are unavailable for opaque models.

3) Training Speed vs. Inference Efficiency: As shown in Fig. 4, KANs are marginally slower to train than MLPs, though most architectures fall within a narrow 8.5–9.6 s/episode range because the RAN simulation dominates wall-clock cost (DDT-4 is an outlier at $\approx$10.9 s due to its deeper tree). For real-world RAN deployments, *inference speed* is often more critical than training speed. The symbolic policy in Eq. (25) can be executed as a series of simple scalar operations, which is significantly faster and more energy-efficient on constrained hardware than the matrix multiplications required by MLPs. While training is slower, the symbolification phase itself is computationally negligible; in our experiments, extracting the symbolic form (Eq. (25)) from a trained spline-based network took less than 1 second on a standard workstation. For practical scenarios, we recommend offline or shadow-mode training for KANs, leveraging their fast and transparent symbolic form for real-time control.

E. Counterfactuals

Fig. 7 illustrates counterfactual selection for the query ($\text{SINR}=15.6$, $\text{RSRP}=-88.3$, $\theta=15°$), whose proposed action is *hold*. Because actions are incremental tilt adjustments ($-1°$, $0°$, $+1°$), counterfactuals are conditioned on the current tilt: the candidate pool is restricted to states observed at $\theta=15°$ and only SINR and RSRP are treated as mutable features. We display $k=10$ counterfactuals, coloured by their flipped action (red: downtilt, blue: uptilt), drawn from the feasible set (gray) and optimally selected by the validity–proximity–diversity objective. The selected CFs lie near the decision boundaries separating the *hold* region from *downtilt* and *uptilt*, and span distinct modes of the state space, providing diverse actionable alternatives that explain under which signal conditions the policy would deviate from holding the current tilt.

Table IV quantifies the counterfactual quality for the query in Fig. 7, comparing the greedy DiverseCF algorithm (Algorithm 1) against a random baseline that uniformly samples k valid counterfactuals (averaged over 100 draws). Both methods

TABLE III
ARCHITECTURE COMPARISON: PARAMETER COUNTS AND INTERPRETABILITY.

KAN Architectures		
Property	**KAN-8**	**KAN-0**
Actor / Critic params	888±56 / 680	192 / 88
Closed-form expression	✓	✓
C_{eff} (tree nodes)	323±165	59
Nonlinearities	sin, cos, x^2	sin, cos, x^2
Analytically verifiable	✓	✓
MLP Baselines		
Property	**MLP-64**	**MLP-128**
Actor / Critic params	4,611 / 4,481	17,411 / 17,153
Closed-form expression	—	—
C_{eff} (tree nodes)	N/A	N/A
Nonlinearities	ReLU	ReLU
Analytically verifiable	—	—
Property	**MLP-256**	
Actor / Critic params	67,587 / 67,073	
DDT Baselines		
Property	**DDT-3**	**DDT-4**
Actor / Critic params	52 / 36	108 / 76
Extractable rules	✓	✓
C_{eff} (tree nodes)	15	31
Nonlinearities	σ	σ
Analytically verifiable	—	—

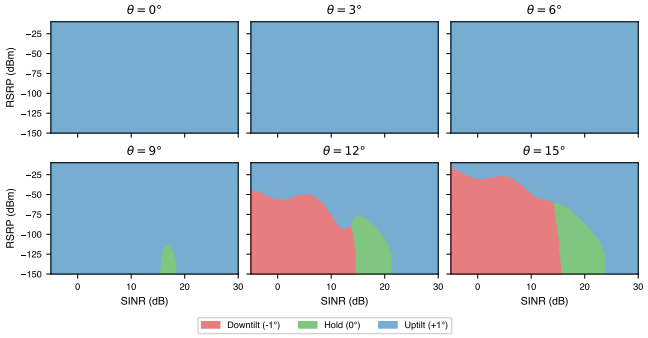


Fig. 6. KAN-8 spline-based policy decision boundaries by tilt θ in the $(\text{SINR}, \text{RSRP})$ plane (red: downtilt, green: hold, blue: uptilt). At low tilts the policy favours uptilt; a hold region appears at $\theta=9°$ for high-SINR, high-RSRP states, and expands with θ. Downtilt emerges at $\theta \geq 12°$ for low-SINR, low-RSRP conditions.

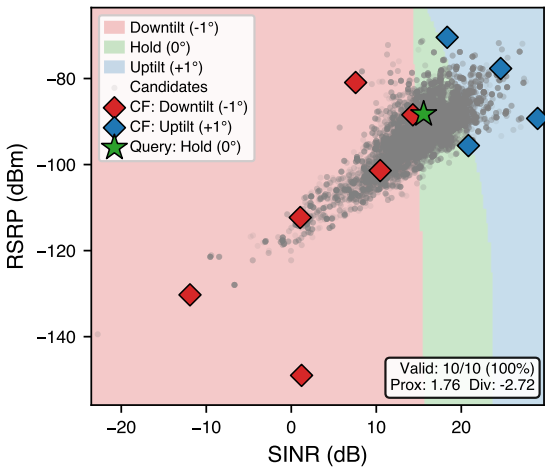


Fig. 7. Counterfactual set (k=10) for the query (SINR=15.6, RSRP=−88.3, θ=15°), conditioned on tilt θ=15°, overlaid on the KAN-8 spline-based policy decision regions. Red/blue diamonds are counterfactuals whose proposed action flips to downtilt/uptilt respectively; the green star marks the query (current action: hold).

TABLE IV
COUNTERFACTUAL METRICS: GREEDY DIVERSECF VS. RANDOM BASELINE (k=10, δ=0.1).

Metric	**DiverseCF**	**Random** (n=100)
Validity (%)	100	100.0 ± 0.0
Proximity ($\bar{d}_M$)	1.76	1.01 ± 0.19
Diversity ($\log\det K$)	-2.72	-8.94 ± 3.47

achieve 100% validity since the candidate pool is pre-filtered to states with a different proposed action. The greedy algorithm substantially improves diversity ($\log\det K = -2.72$ vs. -8.94 ± 3.47), confirming that the joint optimisation objective effectively spreads CFs across distinct regions of the state space, at the cost of slightly higher mean proximity ($\bar{d}_M$=1.76 vs. 1.01 ± 0.19).

VI. CONCLUSIONS

We introduced a method that couples a sparsity- and symbolification-ready KAN actor–critic with a counterfactual generator tailored to closed-form policies. The symbolified KAN exposes analytic logits, enabling exact margins, identification of dominant terms driving class changes, and compact human-readable explanations.

On top of a DiCE-style selector, we implement a rank-1 Cholesky update for the log-determinant diversity term improving the efficiency of the counterfactuals set construction compared to state-of-the-art methods without sacrificing the validity–proximity–diversity trade-off.

We applied the proposed method in a RAN simulation for RET control, showing that the KAN-based controllers matched or slightly exceeded the MLP baselines performance. We highlight that, although we validate the method in a RET use case, it holds generalisation and can be applied into other similar control problems.

We showed that our two-phase KAN training (spline fitting followed by symbolic fine-tuning) retains performance while producing auditable closed-form policies. These symbolic forms enable analytical inspection of decision boundaries and topological properties of the policy surface—capabilities with broad potential for AI verification and risk assessment. The counterfactual explanations, grounded in the symbolic policy, enable experts to transparently assess alternative courses of action.

Current limitations of the method include reliance on a data-driven candidate pool, meaning that the counterfactuals quality directly depends on the observed trajectories. Nonetheless, we intentionally leave the candidates set definition open, so future works could complement this work, by further optimising its definition or even using generative approaches, while remaining feasible, valid, proximal and diverse. A natural extension is to incorporate transition-aware constraints that verify whether a counterfactual state is reachable from the current state under the environment dynamics, bridging the gap between static policy explanations and sequential decision-making.

The action-space has been, for simplicity, limited to discrete actions for now, and therefore future works should incorporate continuous-domain control extensions. Regarding scalability, the sparsity regularisation and pruning steps (Eq. 10) are the primary mechanisms that keep the effective parameter count small as the state-action space grows, enabling KAN-based agents to remain compact and interpretable even in higher-dimensional settings.

REFERENCES

[1] M. K. Shehzad, L. Rose, M. M. Butt, I. Z. Kovacs, M. Assaad, and M. Guizani, "Artificial intelligence for 6g networks: Technology advancement and standardization," *IEEE Vehicular Technology Magazine*, vol. 17, no. 3, p. 16–25, Sep. 2022. [Online]. Available: http://dx.doi.org/10.1109/MVT.2022.3164758

[2] H. Yu, Y. Xiao, J. Wu, Z. He, F. Liu, and J. Liu, "Deep reinforcement learning based beamforming for throughput maximization in ultra-dense networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1021–1026.

[3] F. Vannella, G. Iakovidis, E. A. Hakim, E. Aumayr, and S. Feghhi, "Remote electrical tilt optimization via safe reinforcement learning," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–7.

[4] European Parliament and of the Council, "Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act)," Official Journal of the European Union (OJ L, 12 July 2024), 2024, entered into force 1 August 2024; gradually applicable over 6–36 months. [Online]. Available: https://data.europa.eu/eli/reg/2024/1689/oj

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018. [Online]. Available: https://incompleteideas.net/book/the-book-2nd.html

[6] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12*, 2000.

[7] L. Baird, "Reinforcement learning in continuous time: advantage updating," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 4, 1994, pp. 2448–2453 vol.4.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[9] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," *arXiv preprint arXiv:1502.05477*, 2015. [Online]. Available: https://arxiv.org/abs/1502.05477

[10] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.
[11] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2020. [Online]. Available: https://arxiv.org/abs/1905.07697
[12] "Dice: Diverse counterfactual explanations," https://interpret.ml/DiCE/, accessed 2025-09-25.
[13] V. Bakir, P. Goktas, and S. Akyuz, "Dice-extended: A robust approach to counterfactual explanations in machine learning," *arXiv preprint arXiv:2504.19027*, 2025. [Online]. Available: https://arxiv.org/abs/2504.19027
[14] A. N. Kolmogorov, "On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk SSSR*, vol. 114, pp. 953–956, 1957.
[15] V. I. Arnold, "On functions of three variables," *Dokl. Akad. Nauk SSSR*, vol. 114, no. 4, pp. 679–681, 1957.
[16] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov–arnold networks," 2024.
[17] V. A. Kich *et al.*, "Kolmogorov–arnold network for online reinforcement learning," 2024.
[18] K. Singh, S. Marouani, A. Al Sheikh, P. T. A. Quang, and A. Habrard, "Interpretable reinforcement learning for load balancing using kolmogorov–arnold networks," 2025.
[19] H.-T. Ta, D.-Q. Thai, A. Tran, G. Sidorov, and A. Gelbukh, "Prkan: Parameter-reduced kolmogorov–arnold networks," 2025.
[20] Z. Bozorgasl and H. Chen, "Wav-kan: Wavelet kolmogorov–arnold networks," 2024.
[21] A. Silva, M. Gombolay, T. Killian, I. Jimenez, and S.-H. Son, "Optimization methods for interpretable differentiable decision trees applied to reinforcement learning," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.
[22] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.
[23] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harvard Journal of Law & Technology*, vol. 31, no. 2, pp. 841–887, 2018, arXiv:1711.00399. [Online]. Available: https://arxiv.org/abs/1711.00399
[24] M. Cranmer, "Interpretable machine learning for science with PySR and SymbolicRegression.jl," *arXiv preprint arXiv:2305.01582*, 2023.
[25] F. Ruggeri, A. Terra, R. Inam, and K. H. Johansson, "Evaluation of intrinsic explainable reinforcement learning in remote electrical tilt optimization," in *International Congress on Information and Communication Technology*. Springer, 2023, pp. 835–854.
[26] J. Gallier, "The schur complement and symmetric positive semidefinite (and definite) matrices," University of Pennsylvania, Tech. Rep., 2010. [Online]. Available: https://www.cis.upenn.edu/~jean/gbooks/SchurCompl.pdf

APPENDIX A
KAN REGULARISATION DETAILS

A. Kolmogorov–Arnold Representation Theorem

The architecture of KANs is motivated by the Kolmogorov–Arnold Representation Theorem (KART), which states that any continuous multivariate function $f : [0,1]^n \to \mathbb{R}$ can be written as

$$f(x_1, \dots, x_n) \;=\; \sum_{q=1}^{2n+1} \Phi_q\Big(\sum_{p=1}^{n} \varphi_{q,p}(x_p)\Big) \tag{26}$$

where $\Phi_q : [0,1] \to \mathbb{R}$ are f-independent outer functions and $\varphi_{q,f} : \mathbb{R} \to \mathbb{R}$ are f-dependent inner functions, all continuous [14], [15]. This implies equivalency (not merely approximation) between multivariate and univariate continuity.

B. Sparsity and Pruning Penalties

For a univariate edge function φ evaluated on a batch $\{x^{(s)}\}_{s=1}^{N_p}$, the L_1 activation magnitude is

$$\|\varphi\|_1 \;\equiv\; \frac{1}{N_p} \sum_{s=1}^{N_p} \big|\varphi(x^{(s)})\big| \tag{27}$$

For a layer Φ with n_{in} inputs and n_{out} outputs, the layer-level L_1 norm is $\|\Phi\|_1 \equiv \sum_{i,j} \|\varphi_{i,j}\|_1$. An entropy penalty concentrates activity on fewer edges:

$$\mathcal{H}(\Phi) \;\equiv\; -\sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} \frac{\|\varphi_{i,j}\|_1}{\|\Phi\|_1} \log\Big(\frac{\|\varphi_{i,j}\|_1}{\|\Phi\|_1}\Big) \tag{28}$$

Each φ is implemented as a spline over B-spline bases $\varphi(x) = \sum_{k=1}^{K} c_k\, B_{k,n}(x)$, whose coefficients are regularised as

$$\mathcal{C}(\Phi) \equiv \sum_{k=1}^{K} |c_k|, \quad \mathcal{C}_{\text{diff}}(\Phi) \equiv \sum_{k=1}^{K-1} (c_{k+1} - c_k)^2 \tag{29}$$

encouraging small coefficients and smoother transitions respectively.

APPENDIX B
PROOF OF SECTION III-C3

$\ell_{\text{val}}(c) = 0 \iff y^*(c) \neq y_0$*:* For a given candidate $c \in S_{x_0}$, recall the multi-class margin definition

$$\Delta(c) = \max_{j \neq y_0} a_j(c) - a_{y_0}(c) \tag{30}$$

and the associated validity loss

$$\ell_{\text{val}}(c) = (\delta - \Delta(c))_+ = \max\big(0,\, \delta - \Delta(c)\big) \tag{31}$$

with $\delta > 0$ a fixed margin parameter.

The quantity $\Delta(c)$ represents the advantage of the strongest competing class over the original class y_0. Consequently,

$$\begin{aligned} \Delta(c) > 0 &\iff \exists\, j \neq y_0 :\; a_j(c) > a_{y_0}(c) \\ &\iff y^*(c) = \arg\max_i a_i(c) \neq y_0 \end{aligned} \tag{32}$$

If $\Delta(c) < 0$, then $a_{y_0}(c)$ remains maximal and $y^*(c) = y_0$. From the hinge definition, we have

$$\ell_{\text{val}}(c) = 0 \iff \delta - \Delta(c) \leq 0 \iff \Delta(c) \geq \delta \tag{33}$$

Since $\delta > 0$, the condition $\Delta(c) \geq \delta$ implies $\Delta(c) > 0$, and hence $y^*(c) \neq y_0$. In this case, the loss term vanishes, indicating that the candidate c has successfully altered the model prediction and the new class exceeds the original one by at least the required margin δ.

Therefore,

$$\ell_{\text{val}}(c) = 0 \quad\iff\quad y^*(c) \neq y_0 \text{ and } \Delta(c) \geq \delta \tag{34}$$

That is, the validity loss equals zero precisely when c constitutes a valid counterfactual instance. ∎

APPENDIX C
PROOF OF EQUATION 20

Enlarged Kernel determinant: Let $A := K(C) \in \mathbb{R}^{|C|\times|C|}$ denote the current kernel matrix, and define the similarity vector $d = (K_{ij})_{j\in C} \in \mathbb{R}^{|C|}$. Since $K_{ii} = 1$ by construction, the enlarged kernel can be written in block form as

$$K(C \cup \{i\}) = \begin{bmatrix} A_{|C|\times|C|} & B_{|C|\times 1} \\ C_{1\times|C|} & D_{1\times 1} \end{bmatrix} = \begin{bmatrix} K(C) & d \\ d^T & 1 \end{bmatrix} \quad (35)$$

where $B = d$, $C = d^T$, and $D = 1$.

By the Schur complement [26], we have

$$\det K(C \cup \{i\}) = \det(A)\, \det(D - CA^{-1}B) \quad (36)$$

Because D is scalar, its determinant equals its value, which yields

$$\det K(C \cup \{i\}) = \det K(C)\, \big(1 - d^T K(C)^{-1} d\big) \quad (37)$$

This establishes Equation (20), showing that the determinant of the enlarged kernel decomposes as the determinant of the existing kernel multiplied by the scalar Schur complement term $(1 - d^T K(C)^{-1} d)$. ∎

APPENDIX D
PROOF OF EQUATION 20 SIMPLIFICATION

Simpler determinant: Let $K(C) = LL^T$ be the Cholesky factorisation of the symmetric positive definite kernel, with L lower triangular. Then

$$K(C)^{-1} = (LL^T)^{-1} = L^{-T}L^{-1} \quad (38)$$

Let $u \in \mathbb{R}^{|C|}$ be the unique solution of the triangular system $Lu = d$, so that $u = L^{-1}d$. Substituting into the right hand-side factor of (20)

$$\begin{aligned} d^T K(C)^{-1} d &= d^T L^{-T} L^{-1} d \\ &= (L^{-1}d)^T (L^{-1}d) \\ &= u^T u \end{aligned} \quad (39)$$

Therefore, $1 - d^T K(C)^{-1} d = 1 - u^T u$ ∎

APPENDIX E
PROOF OF EQUATION 21

Rank-1 Cholesky updates: We seek a *lower triangular* Cholesky factor for the enlarged kernel

$$K(C \cup \{i\}) = \begin{bmatrix} K(C) & d \\ d^T & 1 \end{bmatrix}, \quad K(C) = LL^T \quad (40)$$

that is, a matrix of the block form

$$L_{\text{new}} = \begin{bmatrix} L & 0 \\ w^T & \alpha \end{bmatrix}, \quad \text{with } \alpha > 0 \quad (41)$$

such that $L_{\text{new}} L_{\text{new}}^T = K(C \cup \{i\})$. Multiplying out gives

$$L_{\text{new}} L_{\text{new}}^T = \begin{bmatrix} LL^T & Lw \\ w^T L^T & w^T w + \alpha^2 \end{bmatrix} \quad (42)$$

Equating blocks with $K(C \cup \{i\})$ yields the system

$$\begin{aligned} &\text{(i)} \quad LL^T = K(C) \quad \text{(already satisfied)}, \\ &\text{(ii)} \quad Lw = d, \\ &\text{(iii)} \quad w^T w + \alpha^2 = 1. \end{aligned} \quad (43)$$

From (ii) we *define* w as the unique solution of the triangular system $Lw = d$; which can be directly substituted as $u := w = L^{-1}d$. Then, substituting into (iii) gives $\alpha^2 = 1 - u^T u$. We therefore set

$$\gamma := 1 - u^T u, \quad \alpha := \sqrt{\gamma} \quad (44)$$

With this choice we obtain the explicit factor

$$L_{\text{new}} = \begin{bmatrix} L & 0 \\ u^T & \sqrt{\gamma} \end{bmatrix} \quad (45)$$

$$L_{\text{new}} L_{\text{new}}^T = \begin{bmatrix} K(C) & d \\ d^T & 1 \end{bmatrix} = K(C \cup \{i\}) \quad (46)$$

∎

APPENDIX F
PROOF: VANILLA (CUBIC) COMPLEXITY

Proof: Let $m = |C|$. For each candidate i, form $K(C \cup \{i\}) \in \mathbb{R}^{(m+1)\times(m+1)}$ and compute its Cholesky factor. The dense SPD Cholesky cost is

$$T_{\text{chol}}(m+1) = \tfrac{1}{3}(m+1)^3 + O(m^2) = \Theta(m^3). \quad (47)$$

Extracting $\log\det$ from the factor is $O(m)$ and does not change the order. Hence the per-candidate cost is $\Theta(m^3)$ and the per-iteration cost (scoring $n-m$ candidates) is

$$\Theta(n\, m^3). \quad (48)$$

∎

APPENDIX G
PROOF: QUADRATIC COMPLEXITY WITH CHOLESKY UPDATES

Proof: Maintain $K(C) = LL^T$ with L lower triangular. For candidate i, solve $Lu = d$ by forward substitution:

$$T_{\text{solve}}(m) = \sum_{j=1}^{m} j = \tfrac{m(m+1)}{2} = \Theta(m^2). \quad (49)$$

Compute $u^T u$ in $O(m)$. By Section D, $d^T K(C)^{-1} d = u^T u$, and by Section C,

$$\Delta_{\text{div}}(i \mid C) = \log\big(1 - d^T K(C)^{-1} d\big) = \log(1 - u^T u). \quad (50)$$

The rank-1 Cholesky update from Section E appends

$$L_{\text{new}} = \begin{bmatrix} L & 0 \\ u^T & \sqrt{1 - u^T u} \end{bmatrix}, \qquad L_{\text{new}} L_{\text{new}}^T = \begin{bmatrix} K(C) & d \\ d^T & 1 \end{bmatrix}. \quad (51)$$

Thus the per-candidate cost is $\Theta(m^2)$ with $O(m)$ extra working memory (vectors d and u). ∎

APPENDIX H
PROOF: END-TO-END COSTS AND MEMORY

Proof: At greedy step m:

$$\text{vanilla: } \Theta(n\,m^3), \qquad \text{updates: } \Theta(n\,m^2) + O(m).$$

Summations over $m = 1, \ldots, k$ give

$$\sum_{m=1}^{k} m^3 = \left(\tfrac{k(k+1)}{2}\right)^2 = \Theta(k^4),$$
$$\sum_{m=1}^{k} m^2 = \tfrac{k(k+1)(2k+1)}{6} = \Theta(k^3).$$

Therefore

$$\text{vanilla total: } \Theta(n\,k^4),$$
$$\text{updates total: } \Theta(n\,k^3) + O(k^2) = \Theta(n\,k^3).$$

Memory: vanilla forms $(m+1)\times(m+1)$ per candidate $\Rightarrow$ $O(m^2)$ working memory per candidate. Updates store one $m \times m$ factor L ($O(m^2)$ persistent) and use $O(m)$ temporaries per candidate; no candidate-specific matrices are formed. ■

APPENDIX I
GREEDY-DIVERSECF ALGORITHM

Algorithm 1 Greedy-DiverseCF with Cholesky updates

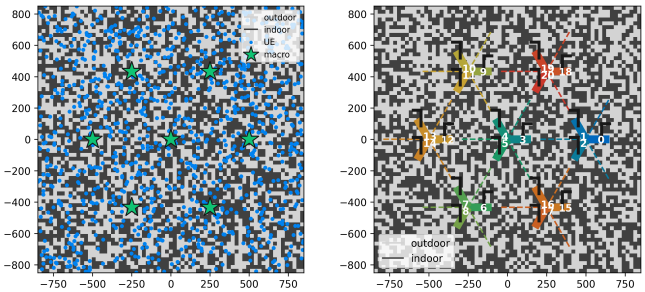
Require: Pool S_{x_0}; losses $\ell_{\text{val}}(c_i)$; CFs k
$C \leftarrow \varnothing$; init. Cholesky factor L
for $m = 1$ to k **do**
  for all $i \notin C$ **do**
    $d_i \leftarrow \left(1/(1+d_M(c_i, c_j))\right)_{j \in C}$
    Compute ΔG_i via Eq. (20)
  end for
  $i^\star \leftarrow \arg\max_i \Delta G_i$
  Solve $Lu = d_{i^\star}$; $\gamma \leftarrow 1 - u^\top u$
  Update L via (21)
  $C \leftarrow C \cup \{i^\star\}$
end for
return C

APPENDIX J
SIMULATED NETWORK LAYOUT

Figures 8a and 8b illustrate the simulated layout and sector geometry. Green markers denote BSs and blue points denote UEs, with a mix of indoor and outdoor locations, obtaining heterogeneous link budgets and traffic demand.

APPENDIX K
CARTPOLE-V1 GENERALIZATION

To assess generalizability beyond the RAN tilt-control task, we evaluate KAN, MLP, and Linear agents on the CartPole-v1 benchmark from OpenAI Gymnasium. CartPole-v1 is a classic continuous-state, discrete-action control task where the agent must balance a pole on a cart by applying left or right forces, receiving a reward of $+1$ per time step up to a maximum of 500.



(a) (b)

Fig. 8. (a) Spatial layout of the simulated RET scenario: base stations and user distribution, (b) Sectorised antenna configuration per BS used in the simulations.

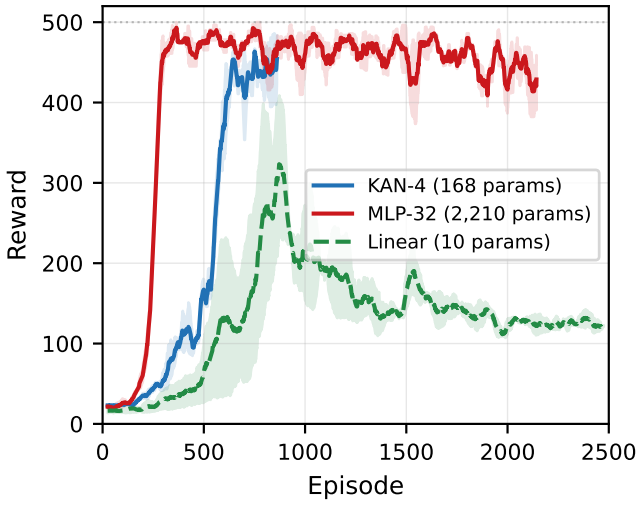


Fig. 9. CartPole-v1 learning curves ($\mu \pm 1\sigma$, 2 seeds). KAN-4 (168 params) converges faster than MLP-32 (2,210 params) to near-optimal reward (500, dotted line).

Setup: All agents use PPO with the same hyperparameters: γ=0.99, λ_{GAE}=0.95, ϵ_{clip}=0.2, learning rate 10^{-3}, and 2,500 training episodes with 2,048-step rollouts. The KAN actor uses grid size 5, spline degree 3, and a single hidden layer of width 4 (denoted KAN-4, 168 parameters). The MLP baseline uses two hidden layers of 32 units each (MLP-32, 2,210 parameters). A single-layer Linear agent (10 parameters) provides a lower baseline.

Results: Fig. 9 shows the learning curves ($\mu \pm 1\sigma$ over 2 seeds, smoothed with a 50-episode window). KAN-4 converges to near-optimal reward ($\approx$475) within $\approx$800 episodes, comparable to MLP-32 which reaches a similar level but takes $\approx$1,500 episodes to stabilize. The Linear agent plateaus at $\approx$130, confirming that CartPole requires nonlinear function approximation for near-optimal control.

Notably, KAN-4 achieves this with 13$\times$ fewer actor parameters (168 vs. 2,210), consistent with the parameter-efficiency advantage observed in the RAN experiments (Sec. V-D). This suggests that the KAN architecture generalizes the compact-

ness and learning-efficiency benefits beyond domain-specific settings.

APPENDIX L
FULL SYMBOLIC POLICY FORMULA

The complete symbolic policy extracted from the KAN-0 agent (seed 42) is:

$$\begin{cases} a_{-1^\circ} & = 0.34\sin(0.96\,\widetilde{s}+5.29) \\ & \quad -1.57\sin(0.52\,\widetilde{\theta}-3.88)+0.68 \\ a_{+0^\circ} & = 0.002\,(9.49\,\widetilde{s}+3.03)^2 \\ & \quad +0.11\sin(0.93\,\widetilde{\theta}+6.83) \\ & \quad -2.44\cos(0.36\,\widetilde{r}+0.35)+1.11 \\ a_{+1^\circ} & = -1.00\sin(0.57\,\widetilde{r}-1.02) \\ & \quad -1.78\cos(0.53\,\widetilde{\theta}-8.46)-0.80 \\ \therefore a^* & = \arg\max\{a_{-1^\circ}, a_{+0^\circ}, a_{+1^\circ}\} \end{cases} \tag{52}$$

where $\widetilde{s}$, $\widetilde{r}$, $\widetilde{\theta}$ denote normalised SINR, RSRP, and tilt (Eqs. (22)–(23)). Each logit is a composition of $\sin$, $\cos$, and x^2 nonlinearities with 7 input references, 28 operators, and 24 numerical coefficients (C_{eff}=59 tree nodes total).