EECE 7268: Final Project Report
Improving Formal Verification Gaps in Neural Network based controllers

Ameya Padwad
(NUID: 002284038)

Risa Samanta
(NUID: 002892447)

December 13, 2025

Abstract

This project examines how architectural and training choices influence the formal verifiability of neural network based controllers for classic control systems. We focus on the Inverted Pendulum and CartPole systems and train compact multilayer perceptron controllers using imitation learning from optimal Linear Quadratic Regulator (LQR) experts. After training, we evaluate controller robustness using sampling based empirical testing and then assess formal guarantees using Interval Bound Propagation (IBP), CROWN, α-CROWN, and Lipschitz based sensitivity analysis. This project observes the effects of using a smoother activation function (Tanh) and pruning of network on these verification bounds. Our results show that Tanh activations consistently produce tighter verification bounds than ReLU across most methods for Pendulum, while the converse is true for CartPole. They also indicate that moderate magnitude pruning substantially tightens bounds for both the systems. The codebase for this project can be found here: GitHub

1 Introduction

Neural networks are increasingly used as controllers for dynamic systems, but providing formal safety guarantees for these controllers remains challenging, which is a necessity to deploy these networks in safety critical environments. Verification tools often produce bounds that are far more conservative than the controller's actual behavior, creating a verification gap. Understanding what design choices cause this gap and how to reduce it is important for making neural controllers reliable in safety critical settings.

In this project, we study this problem using two classical control environments: the Inverted Pendulum and CartPole. For each system, we linearize the dynamics around the upright equilibrium and compute an optimal LQR controller, which serves as the expert policy. We then generate state - action trajectories by simulating the LQR expert from a range of starting states. These trajectories form the supervised dataset used to train the neural network controllers through imitation learning. We then compute verification bounds on these networks using multiple methods, and study the effect of using a smoother activation function and pruning on these bounds.

2 Objective

The objective of this project is to study how architectural and training choices affect the formal verifiability of neural network–based controllers. Specifically, we aim to:

1. Train small neural controllers for the Inverted Pendulum and CartPole systems using imitation learning from optimal LQR experts.
2. Evaluate these controllers using multiple verification approaches, including IBP, CROWN, α-CROWN, and Lipschitz methods.
3. Analyze the verification gap between empirical controller behavior and the guarantees produced by these tools.
4. Investigate techniques such as pruning and smoother activation functions that can reduce this gap while preserving controller performance.

3 Methodology

3.1 Setup

The experiments are conducted on two classical control environments: the Inverted Pendulum and CartPole. For each system, we define a safety region that specifies the range of states under which the controller must keep the system stable.

For the Inverted Pendulum, the state space is 2 dimensional, consisting of the angle θ and angular velocity $\dot{\theta}$. The safe region is defined as:

$$|\theta| \leq 0.5 \text{ rad}, \quad |\dot{\theta}| \leq 1.0 \text{ rad/s},$$

which corresponds to maintaining the pole within approximately 28.6° of the upright position.

For CartPole, the state space is four-dimensional: cart position x, cart velocity $\dot{x}$, pole angle θ, and angular velocity $\dot{\theta}$. The safe region used for verification is:

$$|x| \leq 2.4 \text{ m}, \quad |\theta| \leq 12^\circ,$$

with no hard constraints on velocities, consistent with standard control formulations.

To construct the controllers, we linearize each system around the upright equilibrium and compute an optimal LQR controller. The LQR policy is used to generate state action trajectories across a wide range of initial conditions. For our experiments, we limited the starting state to be within 80% of the safety region for Pendulum, and 30% of the safety region for CartPole. These trajectories form the supervised dataset for training the neural network controllers through imitation learning.

The controllers are implemented as compact multilayer perceptrons. For Inverted Pendulum, the architecture is $2 \rightarrow 16 \rightarrow 8 \rightarrow 1$. For CartPole, the architecture is $4 \rightarrow 32 \rightarrow 16 \rightarrow 1$. Each model is trained using both ReLU and Tanh activations to compare their effect on verifiability. These models are then pruned with varying sparsities using activation pruning and magnitude pruning in an effort to improve the verification bounds.

Evaluation includes empirical robustness testing using output sampling, and formal analyses using IBP, CROWN, α-CROWN, and Lipschitz-based sensitivity bounds.

3.2 Steps to Improve Bounds

3.2.1 Smoother Activation Functions

ReLU is a widely used activation function in neural networks, highly useful because of its piecewise linear nature, which makes it efficient to compute verification bounds on. ReLU typically yields tighter lower bounds across most verification techniques due to this piecewise linear nature. However, as ReLU has an unbounded upper limit, it may result in looser upper bounds in certain situations. ReLU performs exceptionally well in Lipschitz-based verification, where its gradient is either 0 or 1, making Lipschitz constant estimate simple and precise.

Tanh is a smooth bounded activation function that has an output range of [-1, 1]. It automatically limits the output space, which can result in intrinsic limitations when using relaxation-based techniques. Verifiers take advantage of the implicit regularization that this boundedness offers to compute tighter bounds. However, Tanh's highly nonlinear structure makes it more difficult for linear relaxation methods to be precise, which frequently leads to conservative bounds.

3.2.2 Magnitude Pruning

Magnitude pruning removes weights from a neural network whose absolute values fall below a threshold, creating a sparser network with fewer parameters. Magnitude pruning is generally expected to tighten verification bounds by reducing the effective capacity and perturbation propagation through the network. Fewer weights mean smaller accumulated perturbations as they flow forward through layers, which directly translates to tighter certified robustness regions. As sparsity increases, this tightening effect is expected to increase but with diminishing returns.

3.2.3 Activation Pruning

Activation pruning removes neurons whose activations are consistently small across training data, eliminating redundant neurons from the network. Activation pruning is expected to tighten verification bounds more effectively than magnitude pruning because it removes entire neurons rather than individual weights, making the network substantially more sparse, which decreases the propagation of adversarial perturbations. As sparsity increases for activation pruning, the improvement in verification bounds is expected to increase with better scaling than magnitude pruning, since removing inactive neurons has a more direct impact on what the network can compute.

4 Results

The following subsections concisely display comparisons between configurations of the two systems with different activation functions and different pruning methods and sparsities.

4.1 Smoother activation functions

In this project, we have compared Tanh, as a smoother activation function, with ReLU. Tanh, being a smooth bounded activation function, naturally constrains the output space, which can lead to tighter upper bounds in relaxation-based methods like IBP and CROWN. This boundedness provides an implicit regularization that verifiers can use. However, Tanh's non-linear nature makes it difficult for linear relaxation approaches to capture precise boundaries, typically resulting in looser lower bounds.

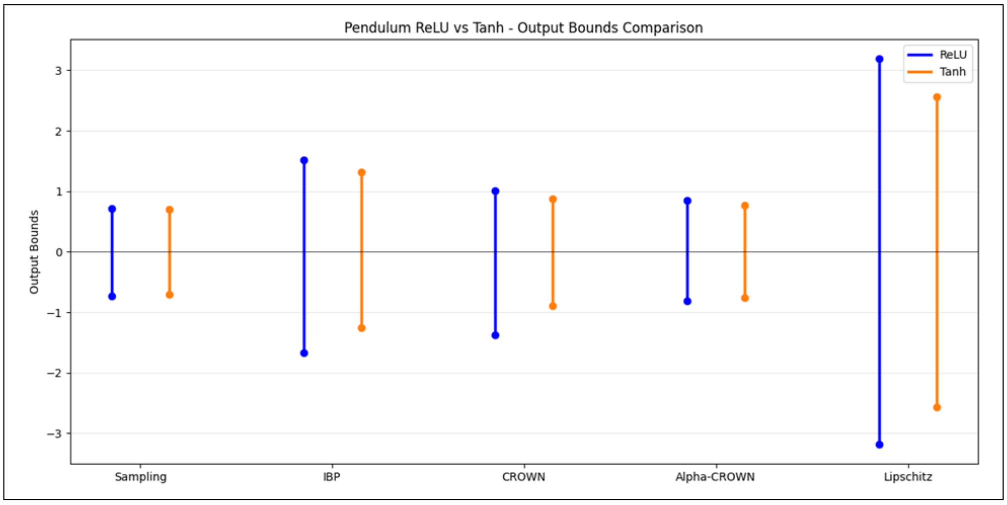

Figure 1: Pendulum - ReLU vs Tanh output bounds comparison

In the Pendulum system (Figure 1), ReLU maintains tighter bounds throughout all verification techniques, although both activations provide comparable bounds for the Sampling and Alpha-CROWN methods. In the Cartpole system (Figure 2), ReLU consistently yields narrower bounds than all other techniques, with the exception of sampling. The Lipschitz constraints (not shown in the figure) are much higher for cartpole for both activation functions (spanning between -150 to +150), although ReLU surpasses Tanh by a large margin. This demonstrates ReLU's ability to generate narrower verification constraints in a complex problem.

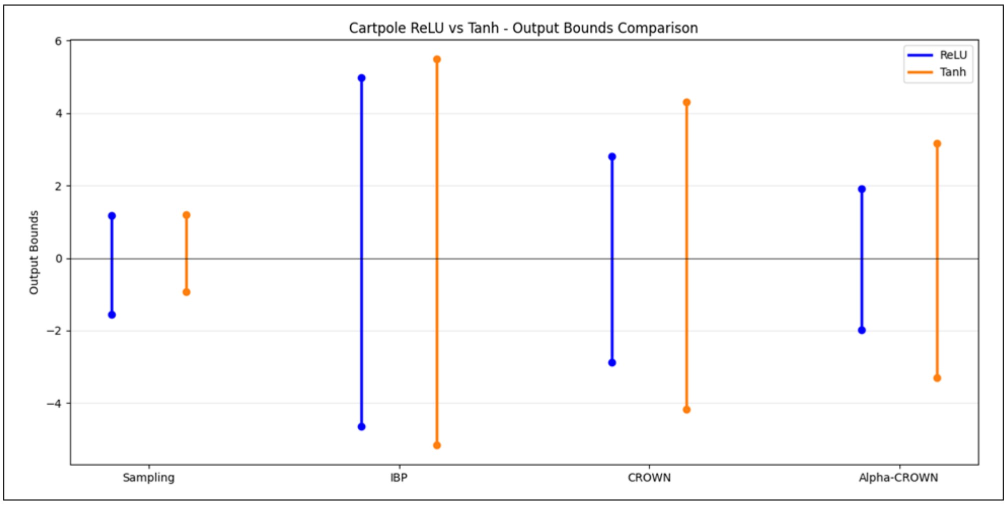

Figure 2: Cartpole - ReLU vs Tanh output bounds comparison

4.2 Pruning Effects

This section portrays the effect of increase in pruning on verification bounds. The following graphs reveal sensitivities of Tanh and ReLU networks to different pruning sparsities across verification methods. The plots display bound width for each verification method as we increase pruning sparsity from left to right. The dotted lines in the graphs represent activation pruning and the solid lines represent magnitude pruning.

4.2.1 Pruning Effects - Pendulum

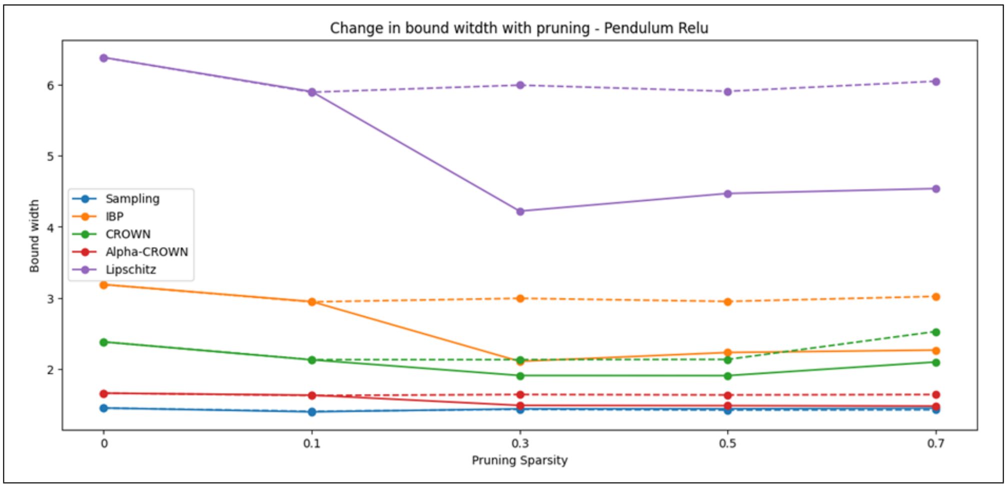

Figure 3: Pendulum Relu - Change in bound width

The ReLU network (Figure 3) exhibits more pronounced bound improvements across both pruning strategies, with Lipschitz bounds declining sharply from 6.4 to approximately 4.2 under magnitude pruning, and IBP bounds decreasing from 3.2 to approximately 2.0. Critically, ReLU networks show that both magnitude and activation pruning contribute meaningfully to bound tightness, with magnitude pruning generally outperforming activation pruning. Another thing to note is the slight increase in bound width for both pruning methods at 0.7 sparsity for CROWN.

In the Pendulum Tanh network (Figure 4), magnitude pruning demonstrates consistent and

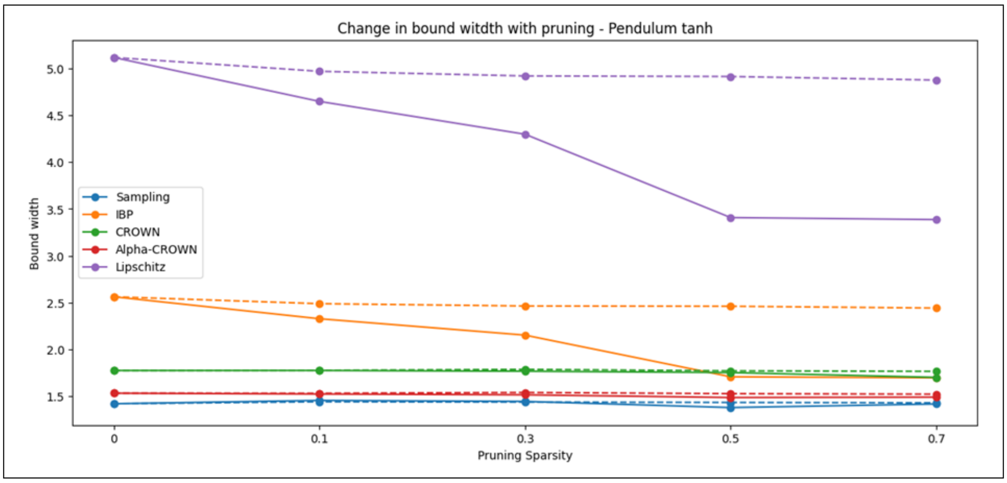

Figure 4: Pendulum Tanh - Change in bound width

substantial improvements in bound width across most methods, while activation pruning provides marginal improvements, suggesting that simply removing inactive neurons is less effective for Tanh networks than structured weight removal. Magnitude pruning sparsity also shows an almost linear relationship with bound width for conservative bound strategies such as IBP and Lipschitz constraints. On the other hand, the tighter boundings strategies seem to have no effect of pruning. This suggests that while Tanh's bounded nature provides some robustness against aggressive pruning, ReLU networks benefit more substantially from pruning operations due to their inherent sparsity properties, making pruning a more effective optimization technique for ReLU-based verified networks.

4.2.2 Pruning Effects - CartPole

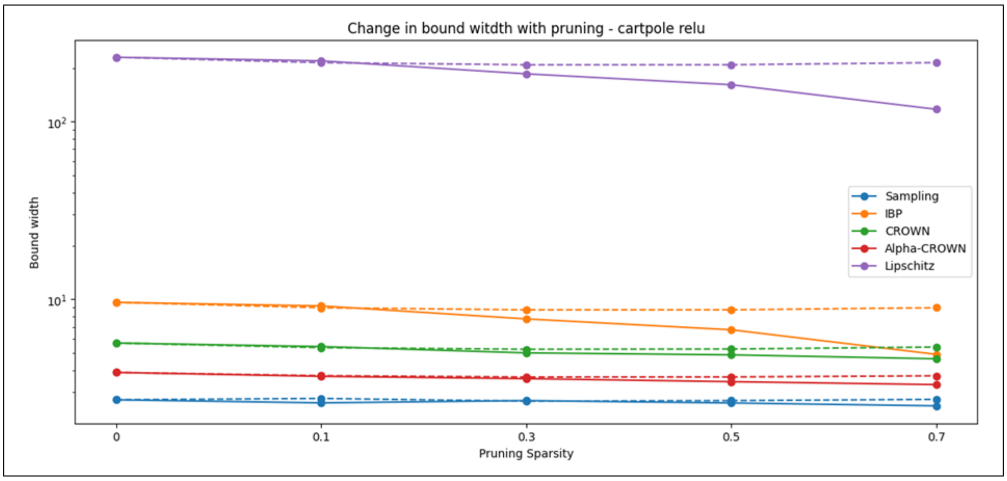

Figure 5: Cartpole Relu - Change in bound width (y-scale log)

The Cartpole pruning analysis (Figures 5 and 6) demonstrates markedly different pruning responsiveness compared to the Pendulum system, with both ReLU and Tanh networks exhibiting remarkable resistance to pruning across all verification methods. In the Cartpole ReLU network (Figure 5), Lipschitz bounds show a notable exception, declining substantially from approximately 150 to 100 as sparsity increases to 0.7, while all relaxation-based methods (IBP, CROWN, Alpha-CROWN) and Sampling remain virtually flat across the pruning sparsity range, indicating minimal sensitivity to network compression. The Cartpole Tanh network (Fig-

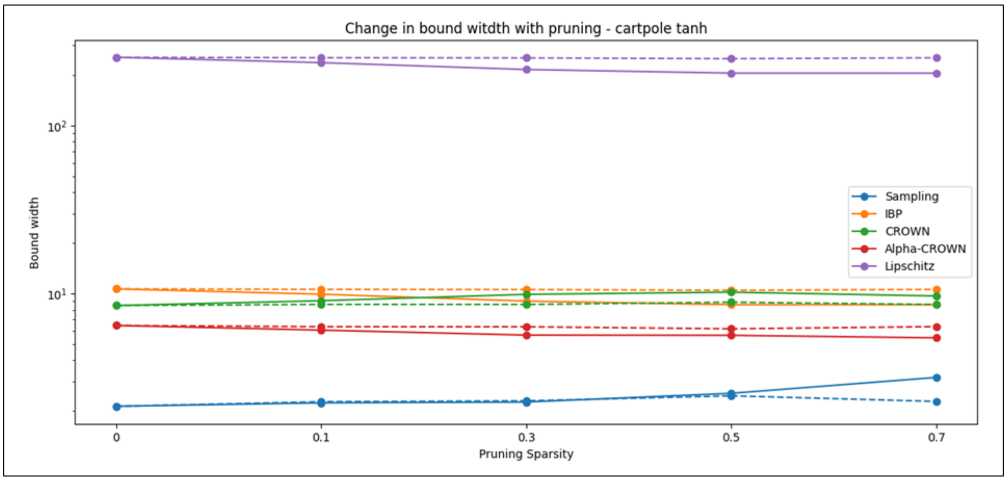

Figure 6: Cartpole Tanh - Change in bound width (y-scale log)

ure 6) displays even more pronounced pruning insensitivity, with nearly all verification methods maintaining constant bound widths regardless of pruning strategy or sparsity level, with the sole exception being a slight increase in Sampling bounds at 0.7 sparsity under magnitude pruning. This stark contrast to the Pendulum results suggests that the verification problem's inherent complexity, as reflected in the substantially larger absolute bound widths for Cartpole (spanning 1-150 compared to Pendulum's 1-6 range), dominates over the effects of network pruning.

4.3 Optimal configurations

Optimizing neural networks for formal verification requires careful consideration of both verification bounds and task accuracy. While tighter verification bounds are essential for proving safety properties, they must not come at the expense of degraded model performance on the actual control task. Optimal configurations must balance these competing objectives, seeking configurations where both metrics remain acceptable within the constraints of the application's safety and performance requirements. This section discusses these optimal configurations that achieve the best verification bounds while maintaining high accuracy.

Metric	**Activation**	**Pruning**	**Value**
Accuracy	Tanh	Magnitude (0.3)	91%
Bounds	Tanh	Magnitude (0.7)	a-CROWN (Bound width: 1.486)
Both	Tanh	Activation (0.5)	Acc: 90%, a-CROWN (Bound width: 1.768)

Table 1: Pendulum Optimal Configurations

Metric	**Activation**	**Pruning**	**Value**
Accuracy	ReLU	Magnitude (0.5)	92%
Bounds	ReLU	Magnitude (0.7)	a-CROWN (Bound width: 3.315)
Both	ReLU	Magnitude (0.5)	Acc: 91%, a-CROWN (Bound width: 3.444)

Table 2: Cartpole Optimal Configurations

The experimental results reveal nuanced trade-offs between these objectives across both systems. In Pendulum, the best accuracy configuration (Tanh with 0.3 magnitude pruning at 91%) differs substantially from the tightest bounds configuration (Tanh with 0.7 magnitude pruning achieving 1.486 bound width via Alpha-CROWN), suggesting aggressive pruning significantly degrades accuracy despite improving verification bounds. The balanced configuration

(Tanh with 0.5 activation pruning at 90% accuracy and 1.768 bound width) demonstrates that a modest 1% accuracy loss yields only a marginal bound degradation, making it a reasonable compromise. The Cartpole results show a more favorable trade-off profile: the optimal accuracy (ReLU with 0.5 magnitude pruning at 92%) and the balanced configuration (same settings at 91% accuracy with 3.444 bound width) exhibit minimal performance variance, suggesting that moderate pruning sparsity provides near-optimal bounds without substantial accuracy sacrifice. Notably, both systems favor moderate pruning levels (0.3-0.5) for balanced configurations, implying that aggressive pruning at high sparsity levels yields diminishing returns on bound improvement while incurring unnecessary accuracy penalties.

4.4 Bounds computation time

The computational efficiency of different verification methods varies significantly depending on activation functions and pruning tactics. As expected for both systems, Sampling and Lipschitz verification perform the fastest computation, followed by IBP and CROWN, while Alpha-CROWN takes significantly longer. The computational cost differential is particularly significant while choosing activation functions. The Pendulum Tanh base configuration takes 11.109s for Alpha-CROWN compared to 5.079s for ReLU, and the Cartpole Tanh base system takes 8.528s for Alpha-CROWN while ReLU takes 5.401s. The results show that ReLU-based networks are computationally more efficient than Tanh networks for verification across all approaches in the base setting.

Pruning dramatically accelerates verification across all methods, with magnitude pruning consistently delivering superior computational speedups compared to activation pruning. In the Pendulum ReLU network, Alpha-CROWN computation time decreases from 5.079s (base) to 1.417s (0.3 sparsity, optimal range) and further to 0.851s (0.7 sparsity). Similarly, Cartpole ReLU Alpha-CROWN time drops from 5.401s (base) to 1.348s (0.5 sparsity). The computational benefits are particularly pronounced for Tanh, being the more complicated nonlinear activation function, as pruning reduces the network's parameter space and thus the complexity of the relaxation computations. For instance, Cartpole Tanh CROWN time decreases from 0.771s (base) to 0.549s (0.5 sparsity), while Alpha-CROWN improves from 5.287s to 2.417s for the same sparsity levels. Activation pruning also accelerates verification, though less consistently than magnitude pruning; Pendulum Tanh Alpha-CROWN time reduces from 11.109s (base) to 6.617s (0.5 sparsity).

5 Conclusion

This project demonstrated that the optimal selection of activation functions and pruning strategies can significantly improve the formal verifiability of neural network-based controllers while maintaining control performance. The following two conclusions can be drawn from this project:

1. **Complexity determines activation function**: Our experiments revealed that Tanh networks consistently produce tighter verification bounds across most methods for Pendulum, while ReLU networks were superior for CartPole.

2. **Moderate pruning can provide substantial improvements in bound tightness and computational efficiency without excessive accuracy degradation**: Magnitude pruning at moderate sparsity levels (0.3-0.5) emerged as a superior optimization technique compared to activation pruning, which significantly improved verification bounds, while maintaining high accuracy and reducing computation cost.