

Supplementary Material — Verification of Neural Control Barrier Functions with Symbolic Derivative Bounds Propagation

Anonymous Author(s)

Affiliation

Address

email

1 A Proofs

2 A.1 Proof of Proposition 1

3 **Lemma 1** (Interval Arithmetic, restated from Section 4.1 in [1]). *For any matrix multiplication*
 4 $\mathbf{A} \cdot \mathbf{x} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, *if \mathbf{x} is entry-wisely bounded as $\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$, i.e. , $\underline{x}_i \leq x_i \leq \bar{x}_i, \forall i = 1 \dots n$, the*
 5 *following inequalities hold for each entry of \mathbf{Ax} ,*

$$[\mathbf{A}]_+ \underline{\mathbf{x}} + [\mathbf{A}]_- \bar{\mathbf{x}} \leq \mathbf{Ax} \leq [\mathbf{A}]_- \underline{\mathbf{x}} + [\mathbf{A}]_+ \bar{\mathbf{x}} \quad (1)$$

6 *where $[\cdot]_+ := \max\{0, \cdot\}, [\cdot]_- := \min\{0, \cdot\}$.*

Proof. For the lower bound $[\mathbf{A}]_+ \underline{\mathbf{x}} + [\mathbf{A}]_- \bar{\mathbf{x}}$, consider the j -th entry of $[\mathbf{Ax}]_j = \sum_{i=1}^n \mathbf{A}_{j,i} x_i$. With the entry-wise bounds $\underline{x}_i \leq x_i \leq \bar{x}_i, \forall i = 1 \dots n$, if $\mathbf{A}_{j,i} \geq 0$, it holds that $\mathbf{A}_{j,i} \underline{x}_i \leq \mathbf{A}_{j,i} x_i$; similarly, if $\mathbf{A}_{j,i} < 0$, it holds that $\mathbf{A}_{j,i} \bar{x}_i \leq \mathbf{A}_{j,i} x_i$. Writing it compactly, we have

$$[\mathbf{A}_{j,i}]_+ \underline{x}_i + [\mathbf{A}_{j,i}]_- \bar{x}_i = \max\{0, \mathbf{A}_{j,i}\} \underline{x}_i + \min\{0, \mathbf{A}_{j,i}\} \bar{x}_i \leq \mathbf{A}_{j,i} x_i$$

. By summing the inequality above over $i = 1, \dots, n$, it holds that

$$[[\mathbf{A}]_+ \underline{\mathbf{x}}]_j + [[\mathbf{A}]_- \bar{\mathbf{x}}]_j = \sum_{i=1}^n [\mathbf{A}_{j,i}]_+ \underline{x}_i + \sum_{i=1}^n [\mathbf{A}_{j,i}]_- \bar{x}_i \leq [\mathbf{Ax}]_j = \sum_{i=1}^n \mathbf{A}_{j,i} x_i$$

7 , which indicates $[\mathbf{A}]_+ \underline{\mathbf{x}} + [\mathbf{A}]_- \bar{\mathbf{x}} \leq \mathbf{Ax}$ holds for each entry j . Similarly, the upper bound $\mathbf{Ax} \leq$
 8 $[\mathbf{A}]_- \underline{\mathbf{x}} + [\mathbf{A}]_+ \bar{\mathbf{x}}$ can be derived in the same way, concluding the proof the interval arithmetic. \square

9 **Proposition 1.** (restated of Proposition 1 in the main text.) *When \mathbf{u} is within a hyper-rectangle*
 10 $\mathcal{U} = [\underline{\mathbf{u}}, \bar{\mathbf{u}}] = \{\mathbf{u} \in \mathbb{R}^m \mid \underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}\}$, *given $\mathbf{x} \in \mathcal{X}$, the minimum value of $\dot{\phi}(\mathbf{x}, \mathbf{u})$ over $\mathbf{u} \in \mathcal{U}$*
 11 *can be found explicitly as,*

$$\min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) = \nabla_{\mathbf{x}} \phi^\top f(\mathbf{x}) + [\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_+ \underline{\mathbf{u}} + [\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_- \bar{\mathbf{u}} = \dot{\phi}(\mathbf{x}, \mathbf{u}_v(\mathbf{x})), \quad (2)$$

12 *where $[\cdot]_+ = \max\{0, \cdot\}, [\cdot]_- = \min\{0, \cdot\}$ and the optimal control input $\mathbf{u}_v(\mathbf{x}) =$*
 13 *$\operatorname{argmin}_{\mathbf{u} \in V(\mathcal{U})} \dot{\phi}(\mathbf{x}, \mathbf{u})$ lies among the vertices $V(\mathcal{U})$ of hyper-rectangle \mathcal{U} given state \mathbf{x} .*

Proof. Based on the chain rule, it holds that

$$\dot{\phi}(\mathbf{x}, \mathbf{u}) = \nabla_{\mathbf{x}} \phi^\top \dot{\mathbf{x}} = \nabla_{\mathbf{x}} \phi^\top f(\mathbf{x}) + \nabla_{\mathbf{x}} \phi^\top g(\mathbf{x}) \mathbf{u}.$$

With $\underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}$, based on Lemma 1, it holds that

$$[\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_+ \underline{\mathbf{u}} + [\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_- \bar{\mathbf{u}} \leq \nabla_{\mathbf{x}} \phi^\top g(\mathbf{x}) \mathbf{u}.$$

Besides, consider the vertices $V(\mathcal{U}) := \{\mathbf{u}_v \mid [\mathbf{u}_v]_i \in \{\underline{u}_i, \bar{u}_i\}, \forall i = 1, \dots, m\}$, we can find the equivalent expression for the lower bounds,

$$[\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_+ \underline{\mathbf{u}} + [\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_- \bar{\mathbf{u}} = \nabla_{\mathbf{x}} \phi^\top g(\mathbf{x}) \mathbf{u}_v(\mathbf{x}),$$

14 *where $[\mathbf{u}_v(\mathbf{x})]_i = \underline{u}_i$ if $[\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_i \geq 0$ and $[\mathbf{u}_v(\mathbf{x})]_i = \bar{u}_i$ if $[\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_i < 0$, showing that*
 15 *there exists $\mathbf{u}_v \in V(\mathcal{U})$ s.t. the lower bound $[\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_+ \underline{\mathbf{u}} + [\nabla_{\mathbf{x}} \phi^\top g(\mathbf{x})]_- \bar{\mathbf{u}}$ can be achieved*
 16 *equivalently, which concludes that proof. \square*

17 A.2 Proof of Theorem 2

18 **Theorem 1.** (restated of Theorem 2 in the main text.) For any control-affine system $h(\mathbf{x}, \mathbf{u}) =$
 19 $f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$ with bounded control input $\mathbf{u} \in \mathcal{U}$, given a learned neural CBF $\phi(\mathbf{x})$ with ReLU
 20 activation functions, suppose the boundary state set $\partial\mathcal{X}_\phi$ is the union of K hyper-rectangles $\Delta\mathcal{X}$ as
 21 $\partial\mathcal{X}_\phi \subset \Delta\mathcal{X}_\phi := \cup_{k=1}^K \Delta\mathcal{X}^{(k)}$, then the following inequality $\max_{\mathbf{x} \in \partial\mathcal{X}_\phi} \min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) + \alpha\phi(\mathbf{x}) \leq$
 22 0 is satisfied if as a sound upper bound of $\nabla_{\mathbf{x}}\phi^\top h(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x})$, the following inequality holds
 23 for any \mathbf{x} in each hyper-rectangle state set $\Delta\mathcal{X} \in \Delta\mathcal{X}_\phi$,

$$[\bar{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_- + [\bar{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_- + \alpha\phi(\mathbf{x}) \leq 0,$$

24 where $[*]_+ = \max\{0, *\} = \text{ReLU}(*)$, $[*]_- = \min\{0, *\} = -\text{ReLU}(-*)$, and $\bar{\mathbf{d}}, \underline{\mathbf{d}}$ and
 25 $\bar{h}(\mathbf{x}, \mathbf{u}_v), \underline{h}(\mathbf{x}, \mathbf{u}_v)$ are the lower and upper bounds of $\nabla_{\mathbf{x}}\phi$ and $h(\mathbf{x}, \mathbf{u}_v)$, respectively.

26 *Proof.* Based on Proposition 1, we have the following inequality hold for any $\Delta\mathcal{X} \in \Delta\mathcal{X}_\phi$,

$$\max_{\mathbf{x} \in \Delta\mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) + \alpha\phi(\mathbf{x}) = \max_{\mathbf{x} \in \Delta\mathcal{X}} \dot{\phi}(\mathbf{x}, \mathbf{u}_v(\mathbf{x})) + \alpha\phi(\mathbf{x}) \leq \max_{\mathbf{x} \in \Delta\mathcal{X}} \dot{\phi}(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}), \quad (3)$$

where \mathbf{u}_v is an approximated constant vertex of optimal control input $\mathbf{u}_v(\mathbf{x})$ for a sound upper bound
 of $\dot{\phi}(\mathbf{x}, \mathbf{u}_v(\mathbf{x}))$ over $\mathbf{x} \in \Delta\mathcal{X}$. Now with the bounded dynamics $\underline{h}(\mathbf{x}, \mathbf{u}_v) \leq h(\mathbf{x}, \mathbf{u}_v) \leq \bar{h}(\mathbf{x}, \mathbf{u}_v)$,
 by Lemma 1, for any $\mathbf{x} \in \Delta\mathcal{X}$ we have

$$\dot{\phi}(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}) = \nabla_{\mathbf{x}}^\top \phi h(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}) \leq [\nabla_{\mathbf{x}}^\top \phi]_- \underline{h}(\mathbf{x}, \mathbf{u}_v) + [\nabla_{\mathbf{x}}^\top \phi]_+ \bar{h}(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}).$$

Besides, with the bounded gradient $\underline{\mathbf{d}} \leq \nabla_{\mathbf{x}}\phi \leq \bar{\mathbf{d}}$, the following inequalities hold

$$[\underline{\mathbf{d}}]_+ \leq [\nabla_{\mathbf{x}}\phi]_+ \leq [\bar{\mathbf{d}}]_+, [\underline{\mathbf{d}}]_- \leq [\nabla_{\mathbf{x}}\phi]_- \leq [\bar{\mathbf{d}}]_-.$$

Then applying Lemma 1 for $[\nabla_{\mathbf{x}}^\top \phi]_- \underline{h}(\mathbf{x}, \mathbf{u}_v)$ and $[\nabla_{\mathbf{x}}^\top \phi]_+ \bar{h}(\mathbf{x}, \mathbf{u}_v)$, we further have the following
 inequality hold for any $\mathbf{x} \in \Delta\mathcal{X}$,

$$\dot{\phi}(\mathbf{x}, \mathbf{u}_v) \leq [\bar{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_- + [\bar{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_-.$$

27 Therefore, if for any \mathbf{x} in each hyper-rectangle state set $\Delta\mathcal{X} \in \Delta\mathcal{X}_\phi$, it holds that

$$[\bar{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_+ [\bar{h}(\mathbf{x}, \mathbf{u}_v)]_- + [\bar{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_+ + [\underline{\mathbf{d}}^\top]_- [\underline{h}(\mathbf{x}, \mathbf{u}_v)]_- + \alpha\phi(\mathbf{x}) \leq 0,$$

and then we have $\dot{\phi}(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}) \leq 0$ for any $\mathbf{x} \in \Delta\mathcal{X}$. Combining Equation (3), we have

$$\max_{\mathbf{x} \in \Delta\mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) + \alpha\phi(\mathbf{x}) \leq \max_{\mathbf{x} \in \Delta\mathcal{X}} \dot{\phi}(\mathbf{x}, \mathbf{u}_v) + \alpha\phi(\mathbf{x}) \leq 0, \forall \Delta\mathcal{X} \in \Delta\mathcal{X}_\phi.$$

Since the exact boundary of $\phi(\mathbf{x}) = 0$ is the subset of all $\Delta\mathcal{X}$, i.e., $\partial\mathcal{X}_\phi \subset \Delta\mathcal{X}_\phi := \cup_{k=1}^K \Delta\mathcal{X}^{(k)}$,
 it holds that

$$\max_{\mathbf{x} \in \partial\mathcal{X}_\phi} \min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) + \alpha\phi(\mathbf{x}) \leq \max_{\Delta\mathcal{X} \in \Delta\mathcal{X}_\phi} \max_{\mathbf{x} \in \Delta\mathcal{X}} \min_{\mathbf{u} \in \mathcal{U}} \dot{\phi}(\mathbf{x}, \mathbf{u}) + \alpha\phi(\mathbf{x}) \leq 0,$$

28 which concludes the proof. \square

29 **Remark 1** (linearly bounded dynamics.). We remark that although the linear bounds of dynamics
 30 $\underline{h}(\mathbf{x}, \mathbf{u}_v), \bar{h}(\mathbf{x}, \mathbf{u}_v)$ can be found through 1-order Taylor models [2, 3, 4] in practice, we can give a
 31 generally valid lower and upper bounds by assuming bounded ℓ_2 operator norm of Hessian matrix
 32 following [5]. For the control-affine system with fixed control input $\mathbf{u}_0, \dot{\mathbf{x}} = h(\mathbf{x}, \mathbf{u}_0)$ with bounded
 33 state $\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$, suppose the ℓ_2 operator norm of Hessian matrix of i -th entry of $h(\mathbf{x}, \mathbf{u}_0)$ is
 34 bounded as $\|\nabla_{\mathbf{x}}^2 h^{(i)}(\mathbf{x}, \mathbf{u}_0)\|_2 \leq M^{(i)}$, then at $\mathbf{x}_0 \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}]$ the following linear bounds can be
 35 found as

$$\underline{h}(\mathbf{x}, \mathbf{u}_0) = \underline{\mathbf{W}}_0 \mathbf{x} + \underline{\mathbf{b}}_0 \leq h(\mathbf{x}, \mathbf{u}_0) \leq \bar{\mathbf{W}}_0 \mathbf{x} + \bar{\mathbf{b}}_0 = \bar{h}(\mathbf{x}, \mathbf{u}_0) \text{ where } \underline{\mathbf{W}}_0 = \bar{\mathbf{W}}_0 = \nabla_{\mathbf{x}}^\top h(\mathbf{x}_0, \mathbf{u}_0),$$

$$\underline{\mathbf{b}}_0^{(i)} = h^{(i)}(\mathbf{x}_0, \mathbf{u}_0) - \nabla_{\mathbf{x}}^\top h^{(i)}(\mathbf{x}_0, \mathbf{u}_0) \mathbf{x}_0 - \frac{1}{2} \|\bar{\mathbf{x}} - \underline{\mathbf{x}}\|_2^2 M^{(i)}, \text{ for } i\text{-th entry of } \underline{\mathbf{b}}_0,$$

$$\bar{\mathbf{b}}_0^{(i)} = h^{(i)}(\mathbf{x}_0, \mathbf{u}_0) - \nabla_{\mathbf{x}}^\top h^{(i)}(\mathbf{x}_0, \mathbf{u}_0) \mathbf{x}_0 + \frac{1}{2} \|\bar{\mathbf{x}} - \underline{\mathbf{x}}\|_2^2 M^{(i)}, \text{ for } i\text{-th entry of } \bar{\mathbf{b}}_0.$$

36 Specifically, if the control-affine system is linear and time-invariant, i.e., $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and $g(\mathbf{x}) = \mathbf{B}$
 37 with constant \mathbf{A}, \mathbf{B} , where the lower and upper bounds will trivially be $\underline{\mathbf{W}}_0 = \bar{\mathbf{W}}_0 = \mathbf{A}, \underline{\mathbf{b}}_0 =$
 38 $\bar{\mathbf{b}}_0 = \mathbf{B}\mathbf{u}_0$.

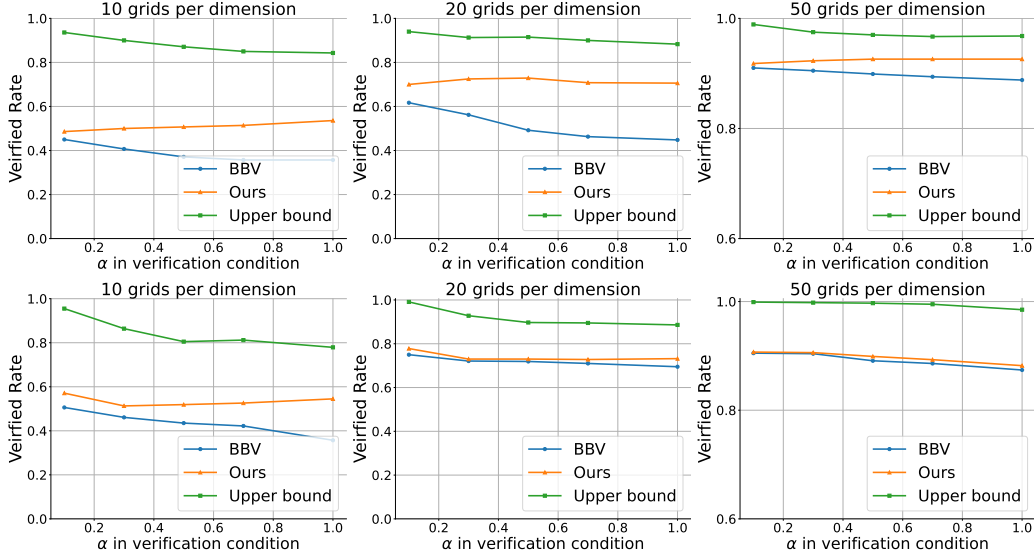


Figure 1: Verified rate with different α in the neural CBF verification condition using different grid sizes (different number of boundary hyper-rectangles) for Dubins Car.

B Experiments

The code can be found in the supplementary material zip file.

B.1 Experiment Environments and Dynamics

All the robot dynamic models are based on the open-sourced package [RobotZoo.jl](#), where Point Robot is modified based on `DoubleIntegrator(D=2)` with zero gravity, Dubins Car is modified based on `DubinsCar` with `radius=0.175`, and Planar Quadrotor is modified based on `PlanarQuadrotor` with `mass=1.0kg`, `gravity=9.81m/s2` and `tip-to-tip distance=0.3m`. Moreover, for the state space, all robots move on a 2D plane within $(0, 4m) \times (0, 4m)$ (horizontal for Point Robot and Dubins Car but vertical for Planar Quadrotor) with a static rectangle obstacle located at the center coordinate $(2m, 1m)$ with sizes of $1m \times 2m$. More specifically, the states of Dubins Car are 2D positions and orientation angle within the 3-dim hyper-rectangle $(0, 4) \times (0, 4) \times (0, \pi)$ and the unsafe states are within 3-dim hyper-rectangle $(1.5, 2.5) \times (0, 2) \times (0, \pi)$. The states of Point Robot are 2D positions and 2D velocities within the 4-dim hyper-rectangle $(0, 4) \times (0, 4) \times (-1, 1) \times (-1, 1)$ and the unsafe states are within 4-dim hyper-rectangle $(1.5, 2.5) \times (0, 2) \times (-1, 1) \times (-1, 1)$. The states of Planar Quadrotor are 2D positions, orientation angle, 2D velocities and angular velocity within the 6-dim hyper-rectangle $(0, 4) \times (0, 4) \times (-0.1, 0.1) \times (-1, 1) \times (-1, 1) \times (-1, 1)$ and the unsafe states are within 6-dim hyper-rectangle $(1.5, 2.5) \times (0, 2) \times (-0.1, 0.1) \times (-1, 1) \times (-1, 1) \times (-1, 1)$. For the control inputs, Dubins Car adopts speed and angular speed within the 2D rectangle $(-1, 1) \times (-1, 1)$. Point Robot adopts 2D accelerations as control input within the 2D rectangle $(-1, 1) \times (-1, 1)$. Planar Quadrotor adopts the thrust forces exerted by the two motors as control input within the 2D rectangle $(4, 6) \times (4, 6)$ to overcome its gravity and move on the vertical plane.

B.2 Implementation Details

Data collection. As shown in the main text, we adopt supervised learning to train the neural CBFs. The data is collected from random trajectories from the safe state space and control input space through the dynamics. To empirically ensure the forward invariance, we discard the second half states and control inputs to avoid the unsafe region of attraction, and only collect the other states and control inputs with the safe labels. To collect unsafe data, we collect random states from the

Number of grids per dimension		10	20	50	100
Regular training	Ours w/o BaB	0.329	0.437	0.875	0.953
	Ours w/ BaB	0.507	0.729	0.926	0.963
Adversarial training	Ours w/o BaB	0.247	0.592	0.837	0.910
	Ours w/ BaB	0.519	0.730	0.899	0.941

Table 1: The ablation study of the branch-and-bound (BaB) in our proposed method under different grid densities with $\alpha = 0.5$ for Dubins Car.

unsafe state space with a similar amount of safe data for balance. To be more detailed, the time of random trajectory is 10s with the time step of 0.1s and the states in the last 5s are omitted. Also, the trajectories that are less than 5s (*e.g.* when the robot collides with obstacles or goes beyond the feasible states) are discarded as well. By repeatedly initializing random states and collecting trajectories with random control inputs through dynamics, we collect 2.5M total pairs of state and control input with 1.4M safe ones for Point Robot, 4.5M total pairs with 2.5M safe ones for Dubins Car and 1.2M total pairs with 0.7M safe ones for Planar Quadrotor as dataset. Then, we randomly choose 10k data as a validation set and use the rest for model training for each robotic dynamics.

Model training. During the model training, we adopt the empirical mean of the positive model predictions as the safe set loss [6] and use the projected gradient descent to find the best-case control input to construct the forward invariance condition loss. To enhance the training efficiency, we only consider the training data along the empirical boundary of $|\phi(\mathbf{x})| < 0.1$ for forward invariance condition loss with $\alpha = 0$. For adversarial training, we adopt project gradient descent over an adjacent cube of each state data to maximize the forward invariance loss, then the gradient descent is based on the worst-case projected states. The size of the adjacent cube in the adversarial training is $1/20$ of each dimension. All the models are trained with Adam for 20 epochs with an initial learning rate of 0.01 and a decay rate of 0.2 every 4 epochs. The neural CBFs for Planar Quadrotor are trained with the weight decay of 0.001.

Verification procedure. The first step of verifying neural CBFs is to find the hyper-rectangles to over-approximate the boundary, *i.e.*, find the superset of all the roots $\phi(\mathbf{x}) = 0$. Assuming the hyper-rectangles are small enough such that the CBF is continuous and monotonic for each dimension of the state, we check all the gridded hyper-rectangles to find if all the vertices give positive or negative CBF values. Based on the mean value theorem and the assumption above, the roots exist in the hyper-rectangles whose vertices give CBF values with opposite signs. Once the hyper-rectangles are obtained along the boundary, we verify the forward invariance condition based on off-the-shelf neural network verification toolboxes [7, 8, 9]. More specifically, given the state specification, we first approximate the optimal control input using one vertex and then find the linear bounds based on [TaylorModels.jl](#) [3]. Then with the Jacobian bounds [10], the condition in Theorem 1 is found to verify if it is no larger than 0. If it does not hold, we adopt branch-and-bound by half-splitting the state specification along each dimension, and conduct the breath-first search to verify Theorem 1 recursively until reaching maximum iteration of 1000. By maintaining all verified sub-specifications, we approximate the optimal control input for other vertex traversals to verify if the union set of all verified sub-specifications equals the whole state specification. All the baselines are conducted in a similar way but with concretized bounds instead of the symbolic one in Theorem 1. We remark that although the procedure may not be the most efficient due to approximating optimal control input, the verification result is sound and the scalability is satisfactory for current robot dynamics. It is marked as future work to make the procedure more efficient for robot dynamics with higher dimensions.

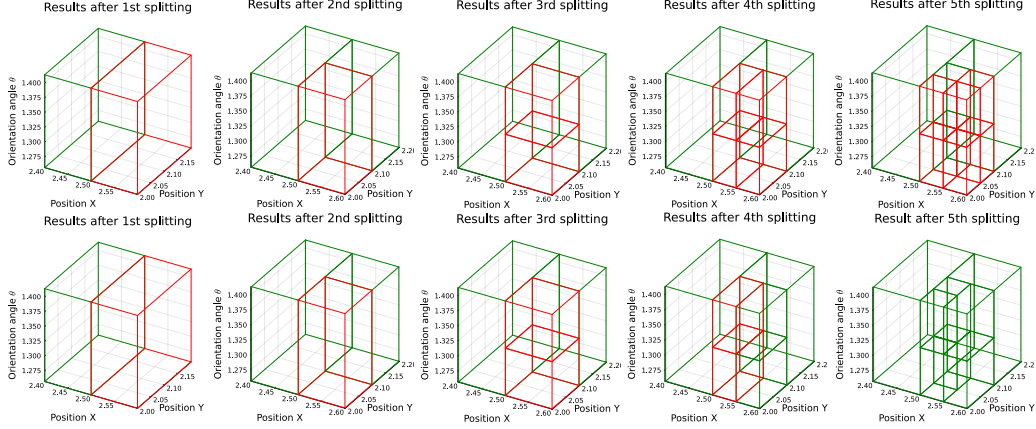


Figure 2: Visualization of branch-and-bound for both baseline BBV [11] (first row) and ours (second row) for Dubins Car with regular-training CBF and $\mathbf{u}_v = [1, 1]$. For each branch after splitting, the green boxes indicate **verified** specifications while the red ones indicate **unknown** specifications.

105 B.3 Additional Results and Analysis

106 **Comparison with different α in the verification condition.** As shown in Figure 1, we compare
 107 our results and the branch-and-bound based baseline BBV [11] with different α in the verification
 108 condition with different grid numbers per dimension. The upper bound indicates how challenging
 109 the verification will be with different α . It can be seen that with larger α , the performance of BBV
 110 decreases more than ours due to larger over-approximation errors. The reason why our results can get
 111 better when α goes up lies in that the approximation errors of gradient bounds through ReLU [10]
 112 become less dominant and the symbolic property becomes more significant. Besides, with more
 113 fine-grained state specifications, the influence of α becomes less because of fewer approximation
 114 errors of both interval arithmetic and ReLU gradient. Adversarial training can also help boost the
 115 verification performance of BBV when the grid size is large, while it can hurt our performance with
 116 small grid sizes due to more training noise during projected gradient descent.

117 **Influence of Branch-and-Bound (BaB) on the proposed method.** Here we conduct an experi-
 118 ment as an ablation study to show the influence of branch-and-bound in the proposed method. As
 119 shown in Table 1, we can find that without branch-and-bound, the performance is significantly re-
 120 duced, especially with fewer boundary hyper-rectangles (larger sizes of grids), showing that branch-
 121 and-bound scheme is essential to the proposed method to alleviate the extra approximation error
 122 caused by finding gradient bounds through ReLU [10].

123 **Visualizaton of branch-and-bound scheme.** From Figure 2, we can see that after each splitting for
 124 the previous unknown specifications, branches will be doubled and the branch-and-bound follows
 125 breadth-first search. With fewer split branches, it can be seen that the coarse specifications cannot
 126 be verified for both base BBV [11] and ours. However, with more splitting, ours can successfully
 127 verify all branches after 5 splits while BBV can only verify some of the finer specifications, leaving
 128 lots of unknown branches to be further split due to looser bounds and larger over-approximation.
 129 The visualization shows that even though with the same approximated optimal control input \mathbf{u}_v ,
 130 ours can give tighter bounds for neural CBF verification with much fewer split times, resulting in a
 131 higher verified rate and shorter verification time.

References

- [1] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404, 2021.
- [2] K. Makino. *Rigorous analysis of nonlinear motion in particle accelerators*. Michigan State University, 1998.
- [3] M. M. Joldes. *Rigorous polynomial approximations and applications*. PhD thesis, Ecole normale supérieure de lyon-ENS LYON, 2011.
- [4] M. Streeter and J. V. Dillon. Automatically bounding the taylor remainder series: Tighter bounds and new applications. *arXiv preprint arXiv:2212.11429*, 2022.
- [5] H. Hu, J. Lan, and C. Liu. Real-time safe control of neural network dynamic models with sound approximation. *arXiv preprint arXiv:2404.13456*, 2024.
- [6] H. Zhang, W. Junlin, V. Yevgeniy, and A. Clark. Exact verification of reLU neural control barrier functions. In *Advances in neural information processing systems*, 2023.
- [7] Neuralverification.jl. <https://github.com/sisl/NeuralVerification.jl>, 2021.
- [8] Modelverification.jl. <https://github.com/intelligent-control-lab/ModelVerification.jl>, 2024.
- [9] α - β -crown. <https://github.com/Verified-Intelligence/alpha-beta-CROWN>, 2022.
- [10] Z. Shi, Y. Wang, H. Zhang, J. Z. Kolter, and C.-J. Hsieh. Efficiently computing local lipschitz constants of neural networks via bound propagation. *Advances in Neural Information Processing Systems*, 35:2350–2364, 2022.
- [11] X. Wang, L. Knoedler, F. B. Mathiesen, and J. Alonso-Mora. Simultaneous synthesis and verification of neural control barrier functions through branch-and-bound verification-in-the-loop training. *arXiv preprint arXiv:2311.10438*, 2023.