VertexCBF: Improving Neural Control Barrier Functions via Vertex-Restricted Control Search

Bojan Derajić[1] · **Sebastian Bernhard**[2] · **Wolfgang Hönig**[1,3]

[1]Technical University of Berlin [2]Technical University of Applied Sciences Augsburg [3]Robotics Institute Germany

arxiv.org/abs/2609.12831

bojan-derajic.github.io/vertexcbf-page

Paper Code & page

01 Motivation

A **CBF-QP safety filter** keeps a robot inside the constraint set $\mathcal{C} = \{x : c(x) \geq 0\}$ by modifying a nominal control input:

$$u^* = \arg\min_{u \in \mathcal{U}} \ \tfrac{1}{2}\|u - u_{\text{nom}}\|^2 \quad \text{s.t.} \ \ L_f h(x) + L_g h(x)\, u + \alpha\big(h(x)\big) \geq 0$$

- Under **input limits**, valid CBFs are hard to design and often **overly conservative**.
- **Grid-based HJ reachability** is exact but scales **exponentially** with the state dimension.
- **PDE-only neural CBFs** tend to converge to **trivial solutions**.

02 The maximal safe set

For dynamics $\dot{x} = \phi(x,u) = f(x) + g(x)u$ with a convex input polytope $\mathcal{U}$, the HJ safety value function

$$V(x) = \sup_{u(\cdot) \in \mathcal{U}} \ \inf_{t \geq 0} \ c\big(x(t)\big)$$

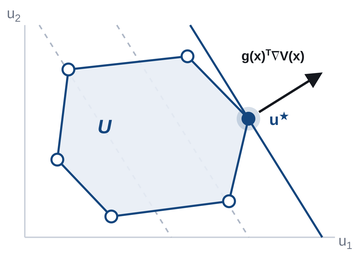
defines the **maximal safe set** $\mathcal{S} = \{x : V(x) \geq 0\}$, the largest controlled invariant subset of $\mathcal{C}$.

→ **We learn a neural CBF V_Θ whose safe set approximates $\mathcal{S}$.**

03 Neural CBF from the HJB equation

V_Θ is trained on the **stationary HJB variational inequality**. The Hamiltonian is linear in u, so its maximum is attained at a vertex of $\mathcal{U}$:

$$\min\Big\{ c(x) - V(x),\ \max_{u \in \mathcal{V}} \nabla V(x)^\top \phi(x,u) \Big\} = 0$$



A linear objective over $\mathcal{U}$ is maximized at a vertex.

04 Model architecture and training

V_Θ is parametrized as a **non-negative residual** subtracted from the constraint:

$$V_\Theta(x) = c(x) - r_\Theta(x), \qquad r_\Theta(x) \geq 0$$

→ **The learned safe set always lies inside $\mathcal{C}$.**

Training objective:

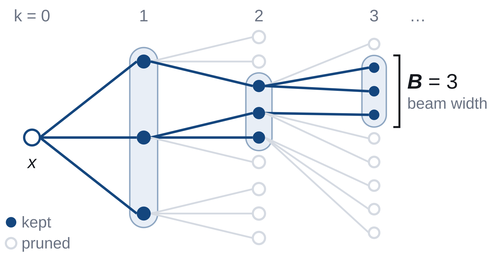
$$\mathcal{L}(\Theta) = \lambda\, \mathcal{L}_{\text{PDE}}(\Theta) + (1-\lambda)\, \mathcal{L}_{\text{data}}(\Theta)$$

- $\mathcal{L}_{\text{PDE}}$: HJB residual on unlabelled states
- $\mathcal{L}_{\text{data}}$: regression on vertex-search labels

05 Vertex-restricted control search

Labels come from a finite-horizon search over sequences of vertex controls:

$$V_{\Delta t}^{K,B}(x) = \max_{u_0,\ldots,u_{K-1} \in \mathcal{V}} \ \min_{0 \leq k \leq K} \ c(x_k)$$



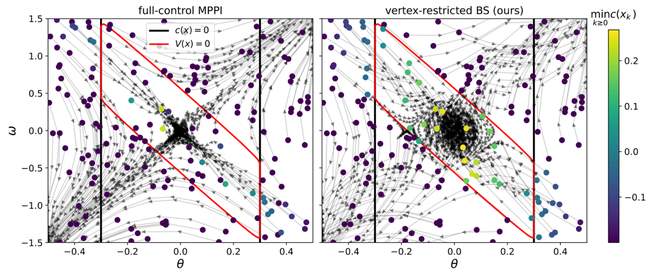
- **Beam search** keeps the best B partial trajectories at each of the K steps.
- **GPU-parallel** over states, beams and vertices; no PDE solver is needed.

06 Evaluation metrics

Predictions are checked with closed-loop rollouts:

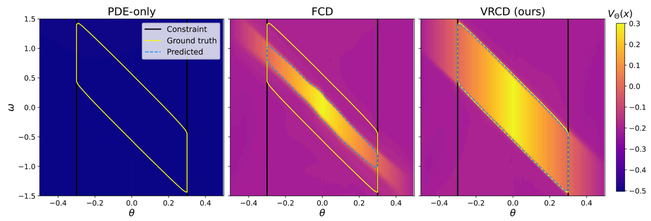
- ρ_{FS} **false-safe**: predicted safe, but the rollout leaves $\mathcal{C}$
- ρ_{FU} **false-unsafe**: predicted unsafe, but the rollout stays safe
- η_{eff} **effective safe volume**: ratio of states predicted and validated safe

07 Better supervision data



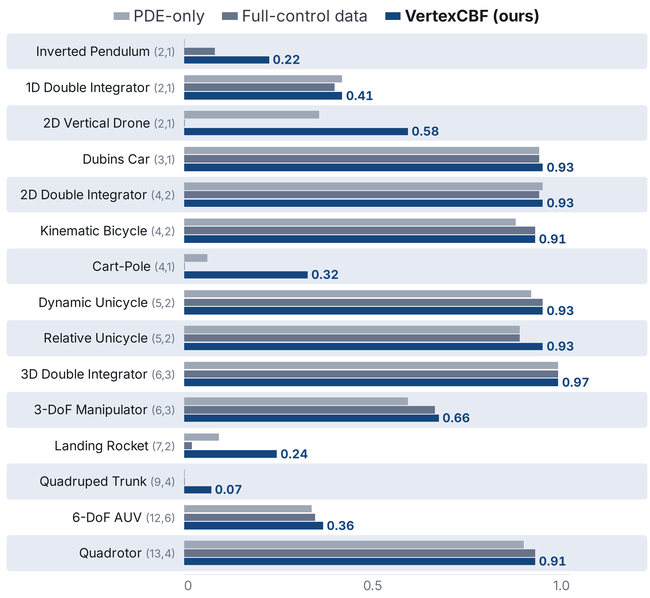
Rollouts from 160 initial states: **full-control MPPI** (left) stays safe from only a few, **vertex beam search** (right) from all states of the safe set.

08 Recovering the maximal safe set



Inverted pendulum: **PDE-only** gives an empty safe set, **full-control data** a conservative one, and **vertex data** (ours) closely approximates the ground truth.

09 Results on 15 systems ($\eta_{\text{eff}} = \frac{\text{vol}(\hat{\mathcal{S}})}{\text{vol}(\mathcal{X})}(1 - \rho_{\text{FS}})$)



Mean of five seeds; (n_x, n_u) in brackets.

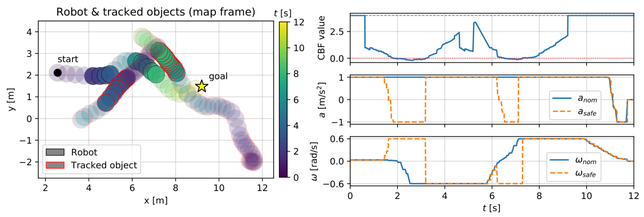
15/15 systems with the highest or tied-highest effective safe volume

≤ 1.2 % false-safe rate on every system; baselines exceed 20 % or return an empty safe set on six

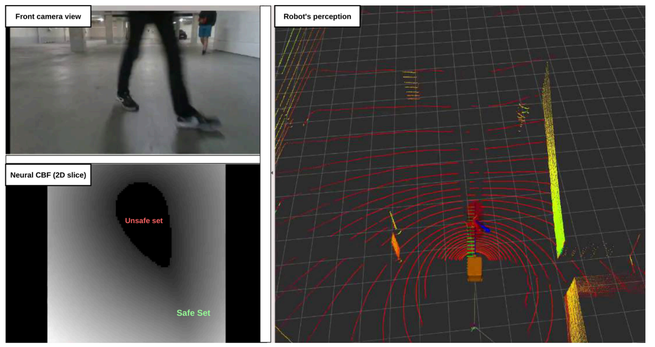
96.4 % IoU with the true safe set on the inverted pendulum, vs. 33.5 % and 1.5 % for the baselines

10 Hardware experiments

- A mobile robot follows an MPC planner, and the learned CBF filters its controls in real time using live pedestrian detections.
- One or two pedestrians cross the robot's path in each trial; **all 10 trials were collision-free**.



One trial: robot and pedestrian paths (left); CBF value and nominal vs. filtered controls (right).



Onboard view: front camera, a 2-D slice of the learned CBF, and the robot's perception.

11 Conclusion

- Searching only over the vertices of $\mathcal{U}$ provides effective supervision without a PDE solver.
- Limitations: no formal guarantees yet; the search cost grows with $|\mathcal{V}|$ and K.
- Future work: error guarantees, systems with many control vertices, and uncertain dynamics.