PoSafeNet: Structured Safety Learning via Compositional Projection

Kiwan Wong[1], **Wei Xiao**[2,3,1 †] **and Daniela Rus**[1,3]
[1] Computer Science and Artificial Intelligence Lab, MIT
[2] School of Electrical and Electronic Engineering, Nanyang Technological University
[3] M3S, Singapore-MIT Alliance for Research and Technology
kiwan588@mit.edu rus@csail.mit.edu wei.xiao@ntu.edu.sg
† Corresponding Authors

Abstract

Safe robot learning often involves multiple heterogeneous safety constraints that cannot always be satisfied simultaneously. Existing neural safety layers typically treat multi-constraint safety as a numerical optimization problem, enforcing all constraints through a single QP-based projection or relaxing conflicts with slack variables. This hides the semantic question of which constraints may be sacrificed under conflict inside solver geometry, penalty weights, or a fixed total hierarchy. We propose PoSafeNet, a poset-structured composable safety layer that makes these conflict semantics explicit. PoSafeNet encodes admissible safety override relations as a partial order and realizes each admissible execution by composing closed-form projections onto Control Barrier Function (CBF)-induced halfspaces. Across multi-obstacle navigation, constrained manipulation, and vision-based autonomous driving, PoSafeNet improves operational feasibility, computational efficiency, and task performance over Differentiable Quadratic Program (dQP)-based, slack-based, and hierarchical safety layers. Videos are available at https://posafenetanonymous.github.io/.

1 Introduction

Learning-based controllers are increasingly equipped with safety layers that compose task-directed nominal actions with formal safety corrections before execution. A common approach is to embed formal safety certificates, such as CBFs [18, 19], into neural policies so that the output control satisfies state-dependent safety constraints. While effective for individual constraints, this paradigm becomes fundamentally more challenging when multiple constraints exist in the same control system. In realistic robotic settings, safety constraints may arise from different failure modes, physical limits, sensing modalities, and task objectives. These constraints are not always simultaneously satisfiable: avoiding a collision may require leaving the center of a lane, respecting a joint limit may require sacrificing tracking accuracy, and different boundary constraints may induce competing corrections. Thus, the central question is not only how to enforce individual constraints, but how safety constraints should be composed when they conflict.

Most existing neural safety layers treat this composition problem primarily as a numerical optimization problem. dQP [4] and Differentiable Model Predictive Control (dMPC) [5] enforce multiple constraints through a single optimization layer, implicitly asking the solver to find a control satisfying all constraints simultaneously. When such constraints are incompatible or ill-conditioned, the resulting optimization can become infeasible, computationally expensive, or unstable to differentiate through [7, 9]. Slack-based relaxations improve numerical feasibility, but they also allow the learned policy to exploit relaxed constraints, weakening the intended safety semantics. Hierarchical

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

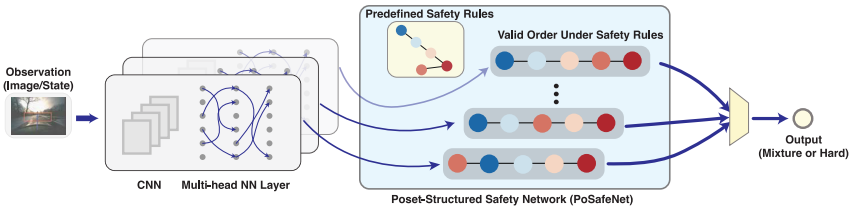


Figure 1: **Overview of PoSafeNet.** The framework replaces QP-based safety layers with multiple closed-form projection heads, each following a valid constraint ordering derived from the given poset.

formulations encode priorities more explicitly, but they typically require a fixed total order over all constraints and still rely on online optimization. These approaches therefore conflate two distinct questions: *what safety semantics should be preserved under conflict*, and *how the corresponding safety correction should be computed*.

We introduce PoSafeNet, a poset-structured composable safety layer that separates these two questions. Rather than assuming all constraints must be enforced symmetrically, PoSafeNet represents admissible safety override relations through a designer-specified Partially Ordered Set (poset) over safety primitives. A constraint cannot be overridden by a strictly lower-priority comparable constraint, while conflicts among incomparable constraints are resolved by selecting one of the poset-consistent admissible executions. Thus, the poset specifies a structural prior over which violations are semantically admissible under conflict.

Given this prior, PoSafeNet realizes admissible executions without solving an online Quadratic Program (QP). For control-affine systems, each CBF constraint induces an affine halfspace in the control input space, and projection onto a single halfspace admits a closed-form solution. PoSafeNet composes these projections sequentially, with each linear extension of the safety poset defining one admissible projection sequence. Each network head implements one such sequence to realize a differentiable, QP-free representation of poset-consistent safety compositions.

We make the following **contributions**:

- We introduce PoSafeNet, a differentiable, composable, and QP-free safety layer for learned controllers. PoSafeNet realizes poset-induced admissible executions by composing closed-form projections onto CBF-induced halfspaces, with each network head corresponding to a linear extension of the safety poset.
- We formulate multi-constraint safe learning as a problem of structured safety composition, where a partial order specifies admissible safety override relations when not all constraints can be simultaneously satisfied.
- We demonstrate that PoSafeNet improves operational feasibility, computational efficiency, and task performance over dQP-based, slack-based, and hierarchical safety layers across multi-obstacle navigation, constrained manipulation, and vision-based autonomous driving.

2 Related Work

Structured and Priority-Based Safety Frameworks. Safety-critical systems have been studied at multiple levels of abstraction. High-level safety frameworks emphasize explicit specifications, critical-scenario identification, world models, and formal verification [20, 8, 6]. These works clarify *what* safety properties should hold, but do not directly address *how* multiple safety requirements should be composed online in continuous control when they conflict.

At the control level, priority-based enforcement has been studied through null-space projection [15, 12] and hierarchical QPs [9], with recent extensions incorporating CBFs into hierarchical safety controllers [10]. These methods resolve conflicts through a fixed lexicographic hierarchy. However, a total hierarchy must rank every pair of constraints, even when no semantic override relation is intended. This can impose arbitrary priorities when safety requirements are heterogeneous, context-dependent, or genuinely incomparable. In contrast, PoSafeNet uses a partial order to encode only the necessary safety override relations under conflict, leaving unrelated constraints incomparable.

Control Barrier Functions and Projection-Based Enforcement. CBFs provide a principled mechanism for enforcing forward invariance by imposing state-dependent constraints on the control input [2, 17]. For control-affine systems, each CBF condition induces an affine halfspace in the control space, commonly enforced through a QP. Although effective in many settings, simultaneous CBF-QP enforcement can face feasibility, scalability, and robustness challenges when constraints are incompatible, heterogeneous, or only partially ordered.

PoSafeNet instead treats the order of CBF-induced corrections as part of the safety-composition structure. Each admissible order is constrained by the safety poset and implemented by closed-form halfspace projections. Thus, projection is not used only as a numerical shortcut, but as a mechanism for realizing poset-structured safety composition.

Safety in Neural Network Controllers. Safe learning often incorporates constraints through penalties or barrier regularization, which provide soft or empirical guarantees [1, 16]. Differentiable optimization layers further embed QPs into neural networks for stronger constraint enforcement [4, 13, 18, 19]. However, these approaches commonly rely on simultaneous enforcement, slack variables, penalty weights, or implicit solver trade-offs to handle conflicts.

In contrast, PoSafeNet models conflict semantics explicitly as a partial order over admissible safety overrides and realizes each admissible execution through poset-consistent sequential projection. By capturing only the necessary priority relations, PoSafeNet avoids artificial symmetry and brittle total-order assumptions while enabling differentiable, QP-free safety composition within neural controllers.

3 Problem Formulation

We consider safe robot learning with *poset-structured safety composition*, where multiple heterogeneous safety constraints may not always be simultaneously satisfiable.

System. Consider a robotic system with state $\boldsymbol{x} \in \mathbb{R}^n$ and control input $\boldsymbol{u} \in \mathbb{R}^m$, governed by

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where f and g are locally Lipschitz continuous.

Nominal controller. We are given a nominal controller $\pi^*(\boldsymbol{x}) = \boldsymbol{u}^*$, which provides supervision for learning but does not necessarily enforce safety.

Safety constraints and override prior. Let $\mathcal{S} = \{1, \dots, N\}$ index safety constraints $b_j : \mathbb{R}^n \to \mathbb{R}$, with constraint j satisfied when $b_j(\boldsymbol{x}) \geq 0$. The constraints are endowed with a partial order $\preceq$, where $i \prec j$ means that constraint j has strictly higher priority than constraint i. Thus, a lower-priority comparable constraint may be sacrificed only to enforce a higher-priority one, while incomparable constraints are resolved by the selected admissible execution. Unlike a total hierarchy, this partial order specifies only necessary override relations and leaves unrelated constraints incomparable. We call $(\mathcal{S}, \preceq)$ a safety poset.

Policy class. We consider a neural policy

$$\pi(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \boldsymbol{u}, \tag{2}$$

where $\boldsymbol{z}$ denotes additional observations or context. Safety is enforced through the policy structure rather than by adding explicit safety constraints to the training loss.

Learning objective. The objective is to minimize the expected imitation loss

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} \left[\ell\big(\pi^*(\boldsymbol{x}), \pi(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\big) \right], \tag{3}$$

while restricting the learned policy to safety corrections whose conflict resolution semantics respect $(\mathcal{S}, \preceq)$.

Definition 3.1 (Poset-Respecting Safety)**.** Let $(\mathcal{S}, \preceq)$ be a safety poset. A safety correction induced by an execution order $\sigma = (j_1, \ldots, j_N)$ is *poset-respecting* if σ is a linear extension of $\preceq$, i.e.,

$$i \prec j \quad \Rightarrow \quad i \text{ appears before } j \text{ in } \sigma.$$

Equivalently, lower-priority comparable constraints are enforced earlier and may be overridden only by higher-priority comparable constraints enforced later.

More generally, if a later projection onto j_ℓ causes an earlier enforced constraint j_k to become violated, the override is admissible only when

$$j_k \prec j_\ell \qquad \text{or} \qquad j_k \parallel j_\ell.$$

Thus, no strictly higher-priority constraint may be overridden by a strictly lower-priority comparable constraint. This is a semantic consistency condition for conflict resolution, rather than a claim that all constraints are simultaneously feasible under conflict.

4 Methodology

We present PoSafeNet, a differentiable safety-composition layer for learned controllers. Given a safety poset, PoSafeNet realizes each admissible override execution as a sequence of closed-form projections onto CBF-induced halfspaces. This replaces simultaneous QP-based enforcement with a structured, QP-free policy parameterization: conflict resolution is specified by the poset and learned over through the network heads, rather than hidden inside solver feasibility, slack variables, or penalty weights.

4.1 Safety Constraints as Halfspace Projections

We first describe how individual safety constraints are operationalized in the control input space. Consider a safety constraint $b_j(\boldsymbol{x}) \geq 0$ as defined in Section 3. For relative-degree-one constraints, the CBF condition $\dot{b}_j(\boldsymbol{x}, \boldsymbol{u}) \geq -\kappa_j b_j(\boldsymbol{x})$ yields an affine inequality in the control input. For higher-relative-degree constraints, we apply the standard high-order CBF (HOCBF) recursion until the control input appears, which yields the same affine halfspace form:

$$A_j(\boldsymbol{x})\, \boldsymbol{u} \;\geq\; c_j(\boldsymbol{x}), \tag{4}$$

where $A_j(\boldsymbol{x}) \in \mathbb{R}^{1\times m}$ and $c_j(\boldsymbol{x}) \in \mathbb{R}$ depend on the system dynamics, the constraint function b_j, and the CBF/HOCBF gains. Details of the CBF/HOCBF-to-halfspace construction are provided in Section B.

Since each CBF constraint induces a *single* affine inequality, the feasible set is a halfspace in the control input space:

$$H_j(\boldsymbol{x}) \;:=\; \big\{\boldsymbol{u} \in \mathbb{R}^m \;\big|\; A_j(\boldsymbol{x})\, \boldsymbol{u} \geq c_j(\boldsymbol{x})\big\}. \tag{5}$$

Safety is enforced by projecting a nominal control input $\boldsymbol{u}_{\text{nom}}$ onto the halfspace $H_j(\boldsymbol{x})$. Specifically, the projection operator

$$\Pi_j(\boldsymbol{u}) \;=\; \arg\min_{\boldsymbol{v}\in\mathbb{R}^m} \|\boldsymbol{v} - \boldsymbol{u}\|^2 \quad \text{s.t. } A_j(\boldsymbol{x})\, \boldsymbol{v} \geq c_j(\boldsymbol{x}) \tag{6}$$

admits the closed-form solution

$$\Pi_j(\boldsymbol{u}) \;=\; \boldsymbol{u} \;+\; \frac{\mathrm{ReLU}\big(c_j(\boldsymbol{x}) - A_j(\boldsymbol{x})\, \boldsymbol{u}\big)}{\|A_j(\boldsymbol{x})\|^2}\, A_j(\boldsymbol{x})^\top. \tag{7}$$

Thus each constraint acts as a differentiable control-space operator: feasible inputs remain unchanged, while infeasible inputs receive the minimal halfspace correction.

4.2 Sequential Safety Projection

While a single safety constraint can be enforced by one projection, multiple constraints are composed by applying projection operators *sequentially*. Let $\sigma = (j_1, j_2, \ldots, j_N)$ denote an ordering of the constraint indices in $\mathcal{S}$. Starting from a nominal control input $\boldsymbol{u}_{\text{nom}}$, define

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}_{\text{nom}}, \qquad \boldsymbol{u}^{(k)} = \Pi_{j_k}\big(\boldsymbol{u}^{(k-1)}\big), \quad k = 1, \ldots, N. \tag{8}$$

Table 1: Scope of the final guarantees.

Constraint	Position in Poset	Final Guarantee
Last-projected maximal constraint	Last layer	Yes
Other maximal constraint – non-interfering	Last layer	Yes
Other maximal constraint – interfering	Last layer	No – poset-respecting
Non-maximal constraint	Earlier layer	No – poset-respecting
All heads share the same last-projected halfspace	Last layer	Yes – unconditional under exact projection and convex mixture weights

The final control input induced by ordering σ is $\boldsymbol{u}_\sigma = \boldsymbol{u}^{(N)}$.

Since later projections can modify the result of earlier ones, later positions in σ have higher effective priority; in particular, the last-enforced constraint cannot be overridden within that execution. Different orderings may therefore induce different safety corrections from the same nominal input, motivating a principled structure over admissible orderings.

4.3 Poset-Structured Safety Composition

We restrict admissible orderings to linear extensions of the safety poset $(\mathcal{S}, \preceq)$: a total order $\sigma = (j_1, \ldots, j_N)$ is admissible if $i \prec j$ implies that i appears before j in σ.

Proposition 4.1 (Poset consistency of linear extensions)**.** *Let $(\mathcal{S}, \preceq)$ be a safety poset. For any linear extension σ of $(\mathcal{S}, \preceq)$, sequential projection according to σ as in* (8) *satisfies poset-respecting safety in the sense of Definition 3.1.*

Proof. If a later projection onto constraint j violates an earlier enforced constraint i, then j appears after i in the linear extension. Hence we cannot have $j \prec i$, since that would require j to appear before i. Therefore the override is possible only when $i \prec j$ or $i \parallel j$, which is exactly the poset-respecting override semantics. □

Proposition 4.1 establishes that every individual linear extension yields a valid safety execution. We now state two formal guarantees: an unconditional result for single-extension execution, and a conditional result for convex mixtures.

Theorem 4.2 (Poset-respecting safety of hard head selection)**.** *Let $\sigma^{(1)}, \ldots, \sigma^{(H)}$ be linear extensions of $(\mathcal{S}, \preceq)$. For any head h, the output $\boldsymbol{u}^{(h)} = \Pi_{\sigma^{(h)}}(\boldsymbol{u}_{\mathrm{nom}}^{(h)})$ satisfies poset-respecting safety for any nominal input and policy parameters. Thus, selecting any single head yields a poset-respecting output without additional assumptions.*

Proof. Immediate from Proposition 4.1. □

Scope of guarantees. Table 1 distinguishes poset-respecting execution from final constraint satisfaction. For a hard-selected linear extension, each exact projection satisfies its own halfspace, but subsequent higher-priority or incomparable projections may invalidate that constraint. The last-projected constraint is maximal and remains satisfied. Other maximal constraints remain satisfied when all subsequent incomparable projections satisfy the non-interference condition in Appendix C, whereas non-maximal constraints may be overridden. Thus, simultaneous satisfaction of all constraints is not guaranteed in general.

For Mixture, simultaneous satisfaction of maximal constraints is guaranteed under the corresponding non-interference conditions in every participating head. If all heads share the same last-projected halfspace, including both its normal and offset, their convex mixture satisfies that halfspace under exact projection, independently of non-interference among earlier projections.

Poset consistency alone does not imply forward invariance. Under the standard CBF assumptions, forward invariance follows for constraints whose CBF conditions remain satisfied by the final control along the trajectory. When multiple maximal constraints are preserved simultaneously, the corresponding intersection of safe sets is forward invariant under these assumptions. No such guarantee is made for overridden non-maximal constraints.

Proposition 4.3 (Conditional safety of convex mixture)**.** *Let $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(H)}$ be the outputs of H heads. If for every constraint $j \in \mathcal{S}$, all head outputs satisfy $\boldsymbol{u}^{(h)} \in H_j(\boldsymbol{x})$, then any convex combination $\boldsymbol{u} = \sum_h \alpha_h \boldsymbol{u}^{(h)}$ with $\alpha_h \geq 0,\ \sum_h \alpha_h = 1$ also satisfies all constraints. More generally, convex mixtures preserve safety when incomparable constraints are non-conflicting, for example when they are not simultaneously active or when projection onto one does not violate the other. Sufficient conditions are given in Appendix C.*

Proof. Each halfspace $H_j(\boldsymbol{x})$ is convex. If $\boldsymbol{u}^{(h)} \in H_j(\boldsymbol{x})$ for all h, then $\sum_h \alpha_h \boldsymbol{u}^{(h)} \in H_j(\boldsymbol{x})$ by convexity. The general non-conflicting case is discussed in Appendix C. □

Non-conflicting incomparable constraints. Incomparable constraints need not be geometrically compatible everywhere; it is sufficient that they do not impose simultaneous conflicting corrections along the considered trajectories. This can occur when their activation regions are disjoint, such as lower- and upper-speed bounds, or when their induced halfspace projections are mutually non-interfering. In such cases, mixing across admissible executions preserves the active safety constraints.

4.4 PoSafeNet Architecture

PoSafeNet is instantiated as a multi-head neural network (Figure 1). Given state $\boldsymbol{x}$ and observations $\boldsymbol{z}$, the network produces H nominal control inputs $\{\boldsymbol{u}_{\text{nom}}^{(h)}\}_{h=1}^{H}$, each associated with a fixed linear extension $\sigma^{(h)}$ of the safety poset. Each head applies the corresponding sequential projection:

$$\boldsymbol{u}^{(h)} \;=\; \Pi_{\sigma^{(h)}}\big(\boldsymbol{u}_{\text{nom}}^{(h)}\big). \tag{9}$$

All heads operate under a shared safety poset; they differ only in their enforcement orderings.

Head combination. The H poset-respecting head outputs are combined via one of two mechanisms:

$$\textit{Soft:}\quad \boldsymbol{u} = \sum_{h=1}^{H} \alpha_h \boldsymbol{u}^{(h)}, \qquad \alpha_h \geq 0,\ \textstyle\sum_h \alpha_h = 1; \qquad\qquad \textit{Hard:}\quad \boldsymbol{u} = \boldsymbol{u}^{(h^*)}, \quad h^* = \arg\max_h \gamma_h, \tag{10}$$

where the mixture weights $\boldsymbol{\alpha}$ or logits $\boldsymbol{\gamma}$ may be state-dependent (produced by an auxiliary network head) or global parameters. During training with hard selection, we use Gumbel-softmax relaxation for differentiability; at inference, h^* is selected deterministically.

Learnable constraint tightness. The gain $\kappa_j \geq 0$ in each CBF condition $\dot{b}_j(\boldsymbol{x}, \boldsymbol{u}) \geq -\kappa_j\, b_j(\boldsymbol{x})$ controls the tightness of constraint j. In PoSafeNet, $\{\kappa_j\}_{j \in \mathcal{S}}$ are treated as learnable parameters optimized jointly with the policy. Learning κ_j allows the model to reduce conservativeness while preserving the affine halfspace structure for any $\kappa_j \geq 0$.

To enforce non-negativity during training, we parameterize each learnable CBF gain as

$$\kappa_j = \text{sigmoid}(\rho_j),$$

where ρ_j is an unconstrained learnable scalar. This guarantees $\kappa_j \in (0, 1)$ throughout training, preserving the affine halfspace structure while preventing negative CBF gains.

Choice of linear extensions. While the number of linear extensions of a poset can be exponential in $|\mathcal{S}|$, PoSafeNet instantiates a finite set of H heads. When $|\mathcal{S}|$ is small, we enumerate all linear extensions; otherwise, we include canonical extensions respecting known semantic hierarchies (e.g., collision avoidance before task comfort).

Control bounds. In the experiments considered in this paper, actuator limits are either encoded as safety constraints in the same halfspace form or the learned nominal controls are parameterized to lie within the admissible actuation range before projection. We do not apply an unmodeled post-hoc clipping step after the PoSafeNet safety layer, since such clipping would itself act as an additional projection and could violate constraints enforced by later, higher-priority projections. When hard box constraints must be imposed, they should be incorporated as additional control-space constraints with an explicitly specified priority in the safety poset, or handled by a final verified safety filter.

Algorithm 1 PoSafeNet: Construction and Inference

Input: Dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$; CBFs $\{b_j \geq 0\}_{j \in \mathcal{S}}$ with poset $(\mathcal{S}, \preceq)$; extensions $\{\sigma^{(h)}\}_{h=1}^{H}$. **Output:** $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z})$.

Compute $\{(A_j(\boldsymbol{x}), c_j(\boldsymbol{x}))\}_{j \in \mathcal{S}}$ via (4)
for $h = 1, \ldots, H$ **do**
  $\boldsymbol{u}^{(h)} \leftarrow \boldsymbol{u}_{\mathrm{nom}}^{(h)}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})$ {nominal}
  for $k = 1, \ldots, N$ **do**
    $\boldsymbol{u}^{(h)} \leftarrow \Pi_{\sigma_k^{(h)}}\big(\boldsymbol{u}^{(h)}; \boldsymbol{x}\big)$ {proj. (7)}
  end for
end for
Combine $\{\boldsymbol{u}^{(h)}\}$ via (10); **return** $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z})$

4.5 Training and Inference

PoSafeNet is trained end-to-end via imitation learning on demonstrations $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}^*) \sim \mathcal{D}$, minimizing

$$\mathcal{L}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}^*) \sim \mathcal{D}}\big[\, \ell\big(\boldsymbol{u}^*,\; \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})\big)\,\big]. \tag{11}$$

Gradients propagate through the closed-form projection operators (7) and the head combination mechanism (10), enabling joint optimization of the nominal policy parameters and the learnable constraint gains $\{\kappa_j\}$. No QP solver is invoked during either training or inference.

Safety guarantee. By Theorem 4.2, hard head selection guarantees poset-respecting safety for any $\boldsymbol{\theta}$ without additional assumptions. Mixture-based execution provides a balanced variant and preserves safety under the compatibility conditions of Proposition 4.3.

The complete construction and inference procedure is summarized in Algorithm 1.

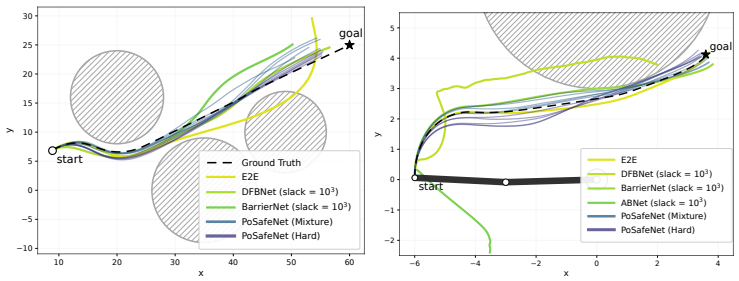
5 Experiments

We evaluate PoSafeNet on three control tasks with increasingly heterogeneous safety structure: multi-obstacle navigation, manipulation with joint-limit precedence, and vision-based autonomous driving. Across these tasks, we test whether PoSafeNet can preserve high-priority safety without solver infeasibility, maintain task performance under conflicting constraints, and reduce the runtime cost of QP-based safety layers.

Benchmark Models. We compare PoSafeNet against baselines spanning end-to-end learning, unstructured safety enforcement, fixed-hierarchy optimization, and our structured safety-composition framework. (1) **End-to-end neural network (E2E)**, a plain neural policy without explicit safety constraints [11, 3]; (2) **Unstructured safety layers**, including BarrierNet [18], ABNet [19], and DFBNet [13], which enforce constraints without an explicit conflict structure, typically through simultaneous constraint enforcement or slack relaxation; (3) **Fixed-hierarchy optimization**, represented by HQP [9], which resolves conflicts through a fixed lexicographic priority order and online optimization; and (4) **Structured safety composition (ours)**, which realizes poset-respecting safety through sequential closed-form projection, using hard selection or mixture over admissible safety executions.

Evaluation metrics. We report the following metrics across tasks. **Op Feas.%**: fraction of timesteps where the safety layer produces a valid output (operationally 100% for PoSafeNet because closed-form projections always return a control). **Obstacle Safety (min)**: minimum obstacle-safety margin over the rollout; non-negative values indicate satisfaction of the obstacle-avoidance constraint. **MSE**: mean squared tracking error against the expert. **FinalDist**: Euclidean distance to the goal at rollout end. **Rollout Time**: average wall-clock time per rollout (seconds). **Top Safety**: whether the highest-priority safety constraint in the task-specific poset is satisfied across all test episodes. Task-specific metrics (e.g., ϕ-violation for joint limits) are defined per subsection. All methods are evaluated under identical settings; full details are in Appendix D.

The slack dilemma. QP-based safety layers face a recurring trade-off: without slack, the QP may become infeasible; with slack, nonzero δ_i relaxes constraint i and allows imitation loss to trade



(a) 2D unicycle navigation (b) Safe manipulation

Figure 2: **Trajectory comparison across tasks under stochastic perturbations.** For PoSafeNet, multiple trajectories arise from different uncertainty realizations, reflecting alternative admissible execution modes. We show a representative rollout here.

Table 2: 2D unicycle obstacle avoidance under noisy input. ABNet (slack $= 10^3$) failed to train and is omitted.

Method	**Primary safety / Feasibility**			Task metrics		Auxiliary
	Op. Feas.% ↑	Obstacle Safety (min) ≥ 0	Top Safety	FinalDist ↓	Rollout time (s) ↓	MSE ↓
E2E [11]	N/A	-81.00	×	26.57	0.028	5.72×10^{-y192}
DFB (slack = 0) [13]	78	-80.78	×	19.43	5.52	8.50×10^{-2}
DFB (slack = 10^3)	100	0.144	✓	15.40	6.57	2.47×10^{-2}
BarrierNet (slack = 0) [18]	44	-43.99	×	7.81	7.09	1.96×10^{-2}
BarrierNet (slack = 10^3)	83	0.00	✓	14.05	7.54	3.27×10^{-2}
ABNet (slack = 0) [19]	35	-61.28	×	13.14	36.37	9.45×10^{-2}
PoSafeNet (Mix)	100	1.63	✓	12.41	1.51	2.92×10^{-2}
PoSafeNet (Hard)	100	0.00	✓	10.95	0.276	2.03×10^{-2}

against safety margin. PoSafeNet avoids this solver-level trade-off by using closed-form projections that always return a control and introduce no slack variables.

5.1 2D Robot Multi-Obstacle Avoidance

A planar unicycle navigates toward a goal while avoiding three circular obstacles (24 test episodes, 100 stochastic rollouts). The three obstacle constraints are treated as mutually incomparable: no obstacle is assigned a fixed priority over another, while simultaneous QP enforcement can still become infeasible under noisy rollouts.

As shown in Table 2, PoSafeNet achieves 100% operational feasibility with non-negative safety margins without relying on slack. PoSafeNet (Hard) attains the best final distance (10.95) among methods that satisfy obstacle safety, while running at 0.276 s per rollout — 27× faster than BarrierNet and 130× faster than ABNet.

5.2 Safe Robot Manipulation under Joint Constraints

For the manipulator task, we evaluate on a two-link planar manipulator with joint limits and obstacle avoidance. The objective is to track a reference end-effector trajectory while strictly satisfying joint-space safety constraints.

Joint-limit constraints must always be satisfied, even at the expense of tracking accuracy, creating a setting with heterogeneous and conflicting requirements. QP-based safety layers handle such constraints either by enforcing them simultaneously or by introducing slack variables, often leading to residual violations or degraded performance under tight joint limits.

In contrast, PoSafeNet enforces constraint precedence through sequential projection, avoiding both infeasibility and slack tuning. Table 3 shows that PoSafeNet is the only method achieving zero ϕ-violation, positive safety margin, while maintaining 100% operationally feasible rollouts. It also achieves the lowest rollout time (0.095 s — 17× faster than BarrierNet) and competitive tracking accuracy, with the mixture variant attaining lower MSE and the hard variant providing faster execution.

Table 3: Benchmark results on the two-link manipulator under noisy input. Metrics are grouped by evaluation priority.

Method	**Primary safety / feasibility**			Secondary safety	Task metrics		Auxiliary
	Op. Feas.% ↑	ϕ-violation (max) ↓	Top Safety	Obstacle Safety (min) ↑	FinalDist ↓	Rollout Time (avg.) ↓	MSE ↓
E2E [11]	N/A	1.98×10^{-1}	×	-12.04	0.702	0.021	9.64×10^{-3}
DFBNet (slack = 0) [13]	100	3.26×10^{-1}	×	-14.99	2.53	1.62	4.04×10^{-1}
DFBNet (slack = 10^3)	100	4.56×10^{-5}	×	-14.84	2.29	1.71	3.86×10^{-1}
BarrierNet (slack = 0) [18]	100	2.51×10^{-1}	×	-9.63	0.672	1.59	1.33×10^{-2}
BarrierNet (slack = 10^3)	100	0.00	✓	-4.07	0.410	1.60	1.21×10^{-2}
ABNet (slack = 0) [19]	76	0.838	×	-14.35	—	0.550	—
ABNet (slack = 10^3)	84	0.837	×	31.03	9.48	0.544	4.20
PoSafeNet (Mix)	100	0.00	✓	0.199	0.361	0.173	9.08×10^{-3}
PoSafeNet (Hard)	100	0.00	✓	0.018	0.527	0.095	1.04×10^{-2}

Table 4: Benchmark results on the VISTA driving task, grouped by constraint hierarchy depth. Results are reported as mean $\pm$ standard deviation over four seeds $\{0, 1, 2, 42\}$.

Method	**Primary safety**		Secondary safety		Task metrics	
	Obstacle Safety (min) ≥ 0	Crash ↓	Lane margin (min) ↑	Time ratio out-of-lane ↓	Progress ↑	$\lvert e_y\rvert$ (mean) ↓
No structured safety layer						
E2E [3]	-43.59 ± 20.89	0.78 ± 0.05	-1.18 ± 1.13	0.052 ± 0.035	37.00 ± 3.18	0.46 ± 0.10
Two-layer hierarchy (obstacle avoidance $\succ$ {L lane, R lane})						
DFBNet [13]	-7.98 ± 5.99	0.75 ± 0.50	1.26 ± 0.85	0.000 ± 0.000	31.44 ± 6.16	0.36 ± 0.62
BarrierNet [18]	-19.26 ± 6.20	0.98 ± 0.05	0.57 ± 0.68	0.000 ± 0.000	42.96 ± 14.84	0.52 ± 0.47
ABNet [19]	-14.29 ± 3.14	0.83 ± 0.24	0.78 ± 0.50	0.000 ± 0.000	45.76 ± 12.86	0.53 ± 0.27
HQP [9]	-0.09 ± 0.00	0.00 ± 0.00	0.06 ± 0.00	0.000 ± 0.000	44.85 ± 0.00	1.07 ± 0.00
PoSafeNet (Hard)	3.37 ± 1.21	0.00 ± 0.00	-0.36 ± 0.19	0.214 ± 0.105	71.39 ± 9.08	1.34 ± 0.08
PoSafeNet (Mixture)	2.42 ± 1.07	0.00 ± 0.00	-0.16 ± 0.06	0.062 ± 0.038	68.78 ± 5.66	0.99 ± 0.15
Four-layer hierarchy (obstacle $\succ$ {L lane, R lane} $\succ$ {speed max, speed min} $\succ$ centerline)						
HQP [9]	0.0002 ± 0.0000	0.00 ± 0.00	1.71 ± 0.00	0.000 ± 0.000	27.14 ± 0.00	0.20 ± 0.00
PoSafeNet (Hard)	3.51 ± 3.88	0.00 ± 0.00	-0.48 ± 0.60	0.193 ± 0.222	75.82 ± 13.72	0.98 ± 0.45
PoSafeNet (Mixture)	1.11 ± 0.52	0.00 ± 0.00	-0.01 ± 0.08	0.001 ± 0.003	61.61 ± 0.17	0.47 ± 0.02

Figure 2b confirms that baselines oscillate or violate joint limits, whereas PoSafeNet tracks the reference while respecting all constraints.

5.3 Vision-Based End-to-End Autonomous Driving

We evaluate PoSafeNet on a driving task in the `VISTA` simulator. The nominal policy maps RGB images to control actions, while the CBF layer uses simulator ground-truth state, lane geometry, and obstacle positions. The task combines obstacle avoidance with lane keeping. The two-layer poset places obstacle avoidance above the incomparable pair {L Lane, R Lane}; the four-layer poset further adds speed regulation and centerline adherence as lower-priority objectives. Table 4 reports mean and standard deviation over four seeds $\{0, 1, 2, 42\}$.

Collision avoidance and task progress. Both PoSafeNet variants and HQP exhibit zero crashes across the evaluated seeds. In contrast, DFBNet, BarrierNet, and ABNet have mean crash rates of 0.75, 0.98, and 0.83, respectively. Zero crashes do not necessarily imply a nonnegative obstacle-barrier margin: two-layer HQP has a reported minimum margin of -0.09 despite its zero crash rate. HQP achieves progress of 44.85 in the two-layer setting and 27.14 in the four-layer setting, with no observed lane departures.

PoSafeNet exposes a safety-performance trade-off. The Mixture variant combines admissible projection-head outputs using learned convex weights. In the two-layer setting, it achieves progress of 68.78 ± 5.66, compared with 44.85 for HQP, while its time-out-of-lane ratio is 0.062 ± 0.038. In the four-layer setting, Mixture achieves progress of 61.61 ± 0.17 and a time-out-of-lane ratio of 0.001 ± 0.003. Relative to two-layer Mixture, its mean absolute lateral deviation decreases from 0.99 to 0.47, accompanied by lower mean progress.

The Hard variant selects one fixed global admissible linear extension at inference. It exhibits higher mean progress than Mixture in both settings, but also higher mean time-out-of-lane ratios and greater across-seed progress variability. These descriptive differences do not establish a reliable cross-seed progress ordering among four-layer Hard, four-layer Mixture, and two-layer Hard. The results demonstrate a trade-off in which lower-priority constraints may yield to support task progress, rather than uniform superiority over HQP across safety and task metrics.

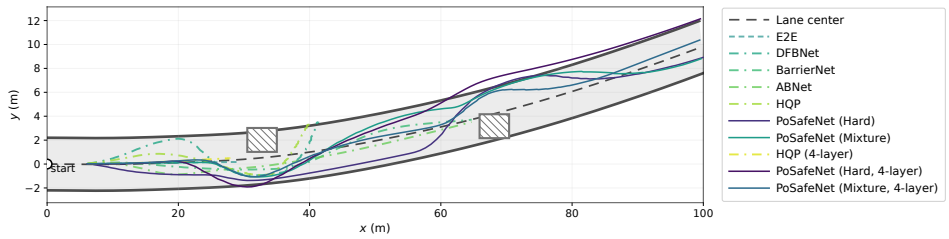


Figure 3: **Representative closed-loop trajectories in VISTA.** Dashed lines denote baselines, while solid lines denote PoSafeNet variants. The displayed trajectories illustrate collisions or limited progress for several baselines, while PoSafeNet avoids obstacle collisions and continues forward. These representative rollouts illustrate the constraint–progress trade-off; aggregate results across four seeds are reported in Table 4.

The comparison with HQP also reflects where the nominal objective enters: HQP introduces nominal-control tracking after fixing the hierarchical relaxation values, whereas PoSafeNet starts each projection head from its nominal control. The progress difference therefore cannot be attributed to the poset structure alone.

6 Conclusions, Limitations and Future Work

We propose PoSafeNet, a learning-based safety-composition framework that models conflict semantics explicitly through partial orders over safety constraints. Rather than assuming that all constraints can always be satisfied simultaneously, PoSafeNet specifies which constraints may override others under conflict and realizes each admissible execution through differentiable closed-form sequential projection. By treating safety composition as a structural property of the policy class, PoSafeNet enables priority-aware, QP-free correction of multiple CBFs while always returning a control output by construction. Across multi-obstacle navigation, constrained robot manipulation, and vision-based autonomous driving, we show that poset-structured safety composition improves operational feasibility, computational efficiency, and task performance compared to unstructured, slack-based, and fixed-hierarchy safety layers.

Limitations and Future Work. PoSafeNet focuses on structured safety-composition problems where task semantics provide a meaningful prior over admissible safety overrides. The CBFs and poset relations are currently specified manually. Test-time insertion of new constraints is structural rather than demonstrated zero-shot generalization: each new constraint requires a CBF/HOCBF representation and specified poset relations, and may require additional heads or retraining. Future work will study how to learn or adapt these posets from demonstrations, interventions, or task data.

Our current implementation uses CBF-induced control halfspaces, requiring the resulting safety conditions to be affine in the optimized control input. The experiments are simulation-based, without physical-robot validation. The safety layer uses simulator ground-truth state and geometry; the observation-noise experiments therefore do not validate the full perception-to-safety-filtering pipeline. Extending the approach to learned barriers, perception-derived constraints, and more general constraint geometries would broaden applicability.

Another direction is to move beyond output-space safety composition and study poset-structured composition in latent policy spaces or parameter spaces with formal guarantees. Finally, we plan to extend the framework to stochastic dynamics, multi-agent interaction, perception uncertainty, and contact-rich manipulation.

Acknowledgments

This work was supported in part by the Singapore MIT Alliance on Research and Technology (SMART) under the Mens, Manus, et Machina (M3S) program, the BARI EMERGE program under grant N00014-26-1-2304, by the MIT-GIST collaboration, Capgemini, The Hong Kong Jockey Club Scholarships, and NTU FA-SUG 025661-00020. We are grateful for this support.

Impact Statement. This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[2] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[3] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2419–2426, 2022. doi: 10.1109/ICRA46639.2022.9812276.

[4] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.

[5] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.

[6] Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry Yershov, Scott Pendleton, James Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks. In *2019 international conference on robotics and automation (ICRA)*, pages 8536–8542. IEEE, 2019.

[7] Jason J Choi, Jasmine Jerry Aloor, Jingqi Li, Maria G Mendoza, Hamsa Balakrishnan, and Claire J Tomlin. Resolving conflicting constraints in multi-agent reinforcement learning with layered safety. *arXiv preprint arXiv:2505.02293*, 2025.

[8] David "davidad" Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, Alessandro Abate, Joe Halpern, Clark Barrett, Ding Zhao, Tan Zhi-Xuan, Jeannette Wing, and Joshua Tenenbaum. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems, 2024. URL `https://arxiv.org/abs/2405.06624`.

[9] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

[10] Jaemin Lee, Jeeseop Kim, and Aaron D. Ames. Hierarchical relaxation of safety-critical controllers: Mitigating contradictory safety conditions with application to quadruped robots. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2384–2391, 2023. doi: 10.1109/IROS55552.2023.10342094.

[11] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[12] Valerio Modugno, Gerard Neumann, Elmar Rueckert, Giuseppe Oriolo, Jan Peters, and Serena Ivaldi. Learning soft task priorities for control of redundant robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 221–226, 2016. doi: 10.1109/ICRA.2016.7487137.

[13] Marcus Pereira, Ziyi Wang, Ioannis Exarchos, and Evangelos Theodorou. Safe optimal control using stochastic barrier functions and deep forward-backward sdes. In *Conference on Robot Learning*, pages 1783–1801. PMLR, 2021.

[14] Alessandro Rucco, Giuseppe Notarstefano, and John Hauser. An efficient minimum-time trajectory generation strategy for two-track car vehicles. *IEEE Transactions on Control Systems Technology*, 23(4):1505–1519, 2015.

[15] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(04):505–518, 2005.

[16] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

[17] Wei Xiao and Calin Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2021.

[18] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.

[19] Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, and Daniela Rus. Abnet: Adaptive explicit-barrier net for safe and scalable robot learning. In *Forty-second International Conference on Machine Learning*, 2025.

[20] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, José Manuel Gaspar Sánchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, Mihai Nica, and Hermann Felbinger. Finding critical scenarios for automated driving systems: A systematic mapping study. *IEEE Transactions on Software Engineering*, 49(3):991–1026, 2023. doi: 10.1109/TSE.2022.3170122.

Appendix of "PoSafeNet: Structured Safety Learning via Compositional Projection"

Table of Contents

A Computation and Training Time **14**

B Control Barrier Function **14**

B.1 From CBF Conditions to Control Halfspaces 14

B.2 Higher-Order CBFs as Control Halfspaces 15

B.3 Safety Constraints as Halfspace Projections 16

C Geometric Conditions for Safe Sequential Projection and Mixing **16**

C.1 CBF Constraints as Control Halfspaces 17

C.2 Two Sources of Compatibility Between CBF Constraints 17

C.2.1 Condition C1: Aligned repulsive directions in configuration space 17

C.2.2 Condition C2: Control-channel isotropy 17

C.3 Implication for Control-Space Halfspace Normals 17

C.4 Sequential Projection and Poset-Structured Safety 18

C.5 Trajectory-Level Non-Interference Audit 19

C.6 Limitations and the Role of Posets 20

D Experimental Details **20**

D.1 2D Robot Multi-Obstacle Avoidance 21

D.2 Safe Robot Manipulation under Joint Constraints 22

D.3 Vision-Based End-to-End Autonomous Driving 23

E Additional Ablations on Projection-Order Sensitivity and Control Quality (VISTA) **25**

E.1 Ablation on the Number of Heads 26

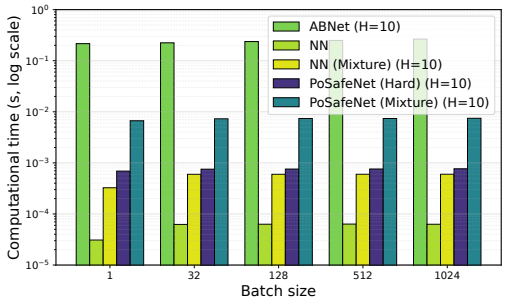
F Four-Level Poset Safety Hierarchy for Heatmap Visualization **27**

F.1 Poset Structure 27

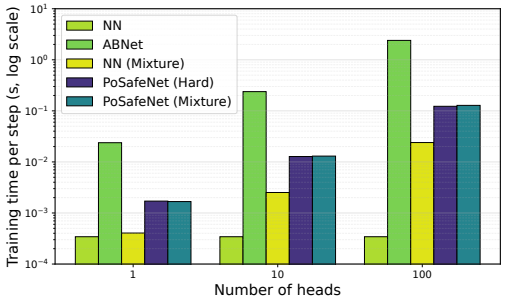
F.2 Additional Barrier Definitions 27

F.3 Interpretation of Heatmaps 28

G Supplementary On Vision Based End-to-End Autonomous Driving **28**



(a) Inference (forward-only) time versus batch size with $H = 10$ (log scale).



(b) Training time per step versus number of heads with $B = 128$ (log scale).

Figure 4: **Computational efficiency of PoSafeNet.** (a) Inference time versus batch size with $H = 10$. (b) Training time per step versus the number of instantiated heads with $B = 128$. With three CBF constraints fixed, PoSafeNet exhibits near-linear timing growth in the instantiated head count and is faster than the evaluated QP-based safety layer. Heads may duplicate admissible extensions; this benchmark does not measure scaling with poset size or the number of distinct linear extensions.

A Computation and Training Time

We compare the computational efficiency of the proposed PoSafeNet with ABNet [19], which enforces safety through a differentiable quadratic programming (dQP) layer [4], as well as with standard neural network baselines. All timing benchmarks are conducted with three simultaneously active CBF constraints, reflecting the target multi-hazard scenarios considered in this work. While ABNet admits a closed-form projection when enforcing at most two linear constraints, this closed-form solution does not generalize to the three-constraint setting evaluated here. Therefore, for a fair and general comparison under multi-constraint settings, we evaluate ABNet using its dQP-based implementation. The dQP layer is implemented using `qpth` from OptNet.

Figure 4a reports inference time under varying batch sizes with a fixed number of heads ($H = 10$), while Figure 4b shows training time per optimization step as the number of heads increases with a fixed batch size of 128. PoSafeNet exhibits computational costs comparable to standard neural networks up to a small constant factor, while significantly outperforming ABNet, whose QP-based safety layer incurs substantially higher overhead.

For timing measurements with $H > |\mathrm{LinExt}(\mathcal{S}, \preceq)|$, we duplicate admissible extensions with independently parameterized nominal heads. This experiment measures computational scaling with the number of instantiated heads, not growth in poset size or the number of distinct linear extensions. The number of admissible extensions can grow combinatorially with poset size. Larger posets therefore require a designer-selected or sampled subset when full enumeration is impractical; the current method does not optimize this subset.

All timing experiments are conducted on a single NVIDIA GPU using PyTorch, with QP layers implemented via `qpth`.

B Control Barrier Function

This appendix clarifies how CBFs and higher-order CBFs induce affine halfspace constraints in the control input space, and provides a closed-form solution to the optimization problem involving a single halfspace constraint.

B.1 From CBF Conditions to Control Halfspaces

Consider a control-affine system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\,\boldsymbol{u}.$$

Given a continuously differentiable control barrier function $b_j(\boldsymbol{x})$, the standard relative-degree-one CBF condition

$$\dot{b}_j(\boldsymbol{x}, \boldsymbol{u}) + \alpha\big(b_j(\boldsymbol{x})\big) \geq 0$$

induces an affine constraint on the control input. Specifically,

$$\dot{b}_j(\boldsymbol{x}, \boldsymbol{u}) = L_f b_j(\boldsymbol{x}) + L_g b_j(\boldsymbol{x})\, \boldsymbol{u},$$

so enforcing the above condition yields

$$A_j(\boldsymbol{x})\, \boldsymbol{u} \;\geq\; c_j(\boldsymbol{x}), \tag{12}$$

where

$$A_j(\boldsymbol{x}) := L_g b_j(\boldsymbol{x}) \in \mathbb{R}^{1\times m}, \qquad c_j(\boldsymbol{x}) := -L_f b_j(\boldsymbol{x}) - \alpha\big(b_j(\boldsymbol{x})\big) \in \mathbb{R}.$$

Thus, although the safety set may be nonlinear in the state space, its CBF condition induces an affine halfspace in the control input space.

B.2 Higher-Order CBFs as Control Halfspaces

Some constraints in our experiments, such as joint limits, lane boundaries, and collision constraints under second-order or vehicle dynamics, have relative degree greater than one with respect to the control input. In these cases, $L_g b_j(\boldsymbol{x}) = 0$, so the standard relative-degree-one CBF condition does not directly constrain $\boldsymbol{u}$. We therefore use higher-order CBFs (HOCBFs), which differentiate the barrier until the control input appears.

For a constraint $b_j(\boldsymbol{x}) \geq 0$ with relative degree r_j, define the recursive functions

$$\psi_{j,0}(\boldsymbol{x}) := b_j(\boldsymbol{x}),$$

and, for $k = 1, \ldots, r_j$,

$$\psi_{j,k}(\boldsymbol{x}) := \dot{\psi}_{j,k-1}(\boldsymbol{x}) + \alpha_{j,k-1}\big(\psi_{j,k-1}(\boldsymbol{x})\big),$$

where each $\alpha_{j,k}$ is an extended class-$\mathcal{K}$ function. The HOCBF condition enforces

$$\psi_{j,r_j}(\boldsymbol{x}, \boldsymbol{u}) \geq 0.$$

Since the system is control-affine and b_j has relative degree r_j, the control input appears linearly in the final derivative. Therefore the HOCBF condition can be written as

$$L_g L_f^{r_j-1} b_j(\boldsymbol{x})\, \boldsymbol{u} \geq -L_f^{r_j} b_j(\boldsymbol{x}) - \eta_j(\boldsymbol{x}), \tag{13}$$

where $\eta_j(\boldsymbol{x})$ collects all lower-order terms generated by the recursive functions $\psi_{j,0}, \ldots, \psi_{j,r_j-1}$ and the class-$\mathcal{K}$ functions. Hence each HOCBF also induces an affine halfspace

$$A_j(\boldsymbol{x})\, \boldsymbol{u} \geq c_j(\boldsymbol{x}), \tag{14}$$

with

$$A_j(\boldsymbol{x}) := L_g L_f^{r_j-1} b_j(\boldsymbol{x}) \in \mathbb{R}^{1\times m}, \qquad c_j(\boldsymbol{x}) := -L_f^{r_j} b_j(\boldsymbol{x}) - \eta_j(\boldsymbol{x}) \in \mathbb{R}.$$

For the common relative-degree-two case, we define

$$\psi_{j,0}(\boldsymbol{x}) := b_j(\boldsymbol{x}), \qquad \psi_{j,1}(\boldsymbol{x}) := \dot{b}_j(\boldsymbol{x}) + \alpha_{j,0}\big(b_j(\boldsymbol{x})\big).$$

The HOCBF condition is

$$\dot{\psi}_{j,1}(\boldsymbol{x}, \boldsymbol{u}) + \alpha_{j,1}\big(\psi_{j,1}(\boldsymbol{x})\big) \geq 0.$$

Expanding this expression gives

$$L_g L_f b_j(\boldsymbol{x})\, \boldsymbol{u} \geq -L_f^2 b_j(\boldsymbol{x}) - L_f \alpha_{j,0}\big(b_j(\boldsymbol{x})\big) - \alpha_{j,1}\big(\psi_{j,1}(\boldsymbol{x})\big), \tag{15}$$

which is again of the form $A_j(\boldsymbol{x})\boldsymbol{u} \;\geq\; c_j(\boldsymbol{x})$. In our implementation, all relative-degree-two constraints are first converted into this affine halfspace form, and the resulting pair $(A_j(\boldsymbol{x}), c_j(\boldsymbol{x}))$ is passed to the same closed-form projection operator as relative-degree-one CBFs.

B.3 Safety Constraints as Halfspace Projections

The affine inequality (12) or (14) defines a halfspace in the control input space. Safety can therefore be enforced by projecting a nominal control input $\boldsymbol{u}_{\mathrm{nom}}$ onto the feasible halfspace associated with constraint j.

Specifically, for a candidate control input $\boldsymbol{u} \in \mathbb{R}^m$, we define the projection operator

$$\Pi_j(\boldsymbol{u}) = \arg\min_{\boldsymbol{v}\in\mathbb{R}^m} \frac{1}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2 \quad \text{s.t.} \quad A_j(\boldsymbol{x})\,\boldsymbol{v} \geq c_j(\boldsymbol{x}). \tag{16}$$

KKT derivation. For notational simplicity, suppress the state dependence and write $A := A_j(\boldsymbol{x})$ and $c := c_j(\boldsymbol{x})$. Consider the projection problem

$$\min_{\boldsymbol{u}\in\mathbb{R}^m} \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{u}_0\|^2 \quad \text{s.t.} \quad A\boldsymbol{u} \geq c,$$

where $\boldsymbol{u}_0$ denotes a given nominal control input.

The Lagrangian is

$$\mathcal{L}(\boldsymbol{u},\lambda) = \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{u}_0\|^2 + \lambda\,(c - A\boldsymbol{u}), \qquad \lambda \geq 0.$$

The Karush–Kuhn–Tucker (KKT) conditions are:

$$\begin{aligned}
\textbf{Stationarity:} \quad & (\boldsymbol{u}-\boldsymbol{u}_0) - \lambda A^\top = \mathbf{0} \;\Rightarrow\; \boldsymbol{u} = \boldsymbol{u}_0 + \lambda A^\top, \\
\textbf{Primal feasibility:} \quad & A\boldsymbol{u} \geq c, \\
\textbf{Dual feasibility:} \quad & \lambda \geq 0, \\
\textbf{Complementary slackness:} \quad & \lambda\,(c - A\boldsymbol{u}) = 0.
\end{aligned}$$

Substituting $\boldsymbol{u} = \boldsymbol{u}_0 + \lambda A^\top$ into the constraint yields

$$c - A(\boldsymbol{u}_0 + \lambda A^\top) = (c - A\boldsymbol{u}_0) - \lambda\|A\|^2,$$

where $\|A\|^2 := AA^\top$ for $A \in \mathbb{R}^{1\times m}$. If $A\boldsymbol{u}_0 \geq c$, complementary slackness implies $\lambda = 0$ and the solution is $\boldsymbol{u}^\star = \boldsymbol{u}_0$. Otherwise, assuming $A \neq 0$, the constraint is active and

$$\lambda^\star = \frac{c - A\boldsymbol{u}_0}{\|A\|^2}.$$

Closed-form solution. Combining both cases, the projection admits the closed-form expression

$$\Pi_j(\boldsymbol{u}) = \boldsymbol{u} + \frac{\mathrm{ReLU}\big(c_j(\boldsymbol{x}) - A_j(\boldsymbol{x})\,\boldsymbol{u}\big)}{\|A_j(\boldsymbol{x})\|^2 + \varepsilon}\, A_j(\boldsymbol{x})^\top. \tag{17}$$

Here $\varepsilon > 0$ is a small numerical stabilizer used in implementation to avoid division by near-zero control authority. When $A_j(\boldsymbol{x}) \neq 0$ and $\varepsilon = 0$, this expression is the exact Euclidean projection onto the halfspace. With small ε, it is a numerically stabilized version of the same closed-form correction.

The ReLU term activates only when the halfspace constraint is violated, ensuring that feasible control inputs remain unchanged while infeasible inputs are corrected toward the constraint boundary. This closed-form operator serves as the basic building block for the sequential projection and mixing framework developed in the main text.

C Geometric Conditions for Safe Sequential Projection and Mixing

This appendix characterizes geometric conditions under which sequential enforcement and convex mixing of multiple CBF constraints preserve feasibility. We also explain why these conditions are frequently satisfied in robotic systems, and how their violation motivates poset-structured safety enforcement.

C.1 CBF Constraints as Control Halfspaces

Consider a control-affine system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\,\boldsymbol{u},$$

and a collection of CBF constraints $b_i(\boldsymbol{x}) \geq 0$. The standard constraint condition

$$\dot{b}_i(\boldsymbol{x}, \boldsymbol{u}) + \alpha\big(b_i(\boldsymbol{x})\big) \geq 0$$

induces an affine constraint on the control input of the form

$$A_i(\boldsymbol{x})\,\boldsymbol{u} \geq c_i(\boldsymbol{x}), \qquad A_i(\boldsymbol{x}) := L_g b_i(\boldsymbol{x}) = \nabla b_i(\boldsymbol{x})^\top g(\boldsymbol{x}) \in \mathbb{R}^{1\times m},$$

which defines a halfspace in the control input space. We write

$$H_i(\boldsymbol{x}) := \{\boldsymbol{u} \in \mathbb{R}^m : A_i(\boldsymbol{x})\boldsymbol{u} \geq c_i(\boldsymbol{x})\}.$$

When the state dependence is clear, we suppress $\boldsymbol{x}$ and write A_i, c_i, H_i. The feasible control set associated with multiple CBF constraints is the intersection of such halfspaces.

C.2 Two Sources of Compatibility Between CBF Constraints

Empirically, sequential projection and convex mixing of CBF constraints often preserve feasibility in robotic systems, though this behavior is not universal. We identify two geometric conditions—acting in different spaces—that explain when such compatibility is expected.

C.2.1 Condition C1: Aligned repulsive directions in configuration space

Each CBF encodes a repulsive constraint with respect to a hazard in the configuration space $\boldsymbol{q}$. For common geometric barriers such as obstacle avoidance, collision radii, or lane boundaries, the gradient $\nabla_{\boldsymbol{q}} b_i(\boldsymbol{x})$ points in the direction that locally increases the safety margin.

We assume that, at a given configuration,

$$\nabla_{\boldsymbol{q}} b_i(\boldsymbol{x})^\top \nabla_{\boldsymbol{q}} b_j(\boldsymbol{x}) \geq 0, \qquad i \neq j,$$

that is, the repulsive directions associated with different hazards are not opposing. Geometrically, this corresponds to hazards whose avoidance induces compatible motions, such as multiple constraints requiring deceleration or steering in the same general direction.

C.2.2 Condition C2: Control-channel isotropy

Let $g_{\boldsymbol{q}}(\boldsymbol{x})$ denote the block of $g(\boldsymbol{x})$ mapping control inputs to configuration dynamics $\dot{\boldsymbol{q}}$. We assume that locally

$$g_{\boldsymbol{q}}(\boldsymbol{x}) g_{\boldsymbol{q}}(\boldsymbol{x})^\top = \gamma(\boldsymbol{x}) I, \qquad \gamma(\boldsymbol{x}) > 0,$$

or more generally that $g_{\boldsymbol{q}} g_{\boldsymbol{q}}^\top$ approximately preserves angles in configuration space. This condition holds exactly for single-integrator models and approximately for many fully actuated or feedback-linearized robotic systems.

C.3 Implication for Control-Space Halfspace Normals

Under Conditions C1–C2, the control-space halfspace normals inherit the alignment of the underlying configuration-space repulsive directions.

Lemma C.1 (Compatibility of CBF halfspaces in control space)**.** *Under Conditions C1–C2, the control-space normals satisfy*

$$A_i(\boldsymbol{x}) A_j(\boldsymbol{x})^\top \geq 0, \qquad \forall\, i \neq j.$$

Proof. Restricting attention to the configuration component, we have

$$A_i(\boldsymbol{x}) = \nabla_{\boldsymbol{q}} b_i(\boldsymbol{x})^\top g_{\boldsymbol{q}}(\boldsymbol{x}) \in \mathbb{R}^{1\times m}.$$

Thus, for any $i \neq j$,

$$\begin{aligned} A_i(\boldsymbol{x}) A_j(\boldsymbol{x})^\top &= \nabla_{\boldsymbol{q}} b_i(\boldsymbol{x})^\top g_{\boldsymbol{q}}(\boldsymbol{x}) g_{\boldsymbol{q}}(\boldsymbol{x})^\top \nabla_{\boldsymbol{q}} b_j(\boldsymbol{x}) \\ &= \gamma(\boldsymbol{x})\, \nabla_{\boldsymbol{q}} b_i(\boldsymbol{x})^\top \nabla_{\boldsymbol{q}} b_j(\boldsymbol{x}), \end{aligned}$$

where the second equality follows from Condition C2. By Condition C1 and $\gamma(\boldsymbol{x}) > 0$, the claim follows. □

C.4 Sequential Projection and Poset-Structured Safety

We now formalize the relationship between sequential projection, the safety poset, and convex mixing across multiple admissible executions.

Theorem C.2 (Poset safety under sequential projection and antichain mixing)**.** *Let $(\mathcal{S}, \preceq)$ be a safety poset. Each constraint $i \in \mathcal{S}$ induces a closed convex feasible set $H_i \subseteq \mathbb{R}^m$. For CBF halfspaces,*

$$H_i = \{\boldsymbol{u} \in \mathbb{R}^m : A_i \boldsymbol{u} \geq c_i\}, \qquad A_i \in \mathbb{R}^{1\times m}.$$

Fix any linear extension $\sigma = (j_1, \ldots, j_{|\mathcal{S}|})$ of $\preceq$, with lower-priority constraints enforced earlier and higher-priority constraints enforced later:

$$i \prec j \quad \Rightarrow \quad i \text{ appears before } j \text{ in } \sigma.$$

Given a nominal control $\boldsymbol{u}_{\mathrm{nom}}$, define the sequential projection iterates

$$\boldsymbol{u}^{(0)} := \boldsymbol{u}_{\mathrm{nom}}, \qquad \boldsymbol{u}^{(t)} := \Pi_{j_t}\big(\boldsymbol{u}^{(t-1)}\big), \quad t = 1, \ldots, |\mathcal{S}|,$$

where Π_j denotes Euclidean projection onto H_j. Let

$$\boldsymbol{u}_{\mathrm{out}} := \boldsymbol{u}^{(|\mathcal{S}|)}.$$

Assume that for every incomparable pair $i \parallel j$, projection onto one constraint does not violate the other:

$$\boldsymbol{u} \in H_j \Rightarrow \Pi_i(\boldsymbol{u}) \in H_j, \qquad \boldsymbol{u} \in H_i \Rightarrow \Pi_j(\boldsymbol{u}) \in H_i. \tag{NI}$$

For halfspaces, a sufficient geometric condition is

$$A_i A_j^\top \geq 0$$

together with nonempty intersection $H_i \cap H_j \neq \emptyset$; Lemma C.1 gives one common setting where this alignment condition arises.

Then the following hold.

1. ***Poset-respecting override.*** *After a constraint i has been enforced, any later projection that violates i can only correspond to a constraint j satisfying either*

   $$i \prec j \qquad \text{or} \qquad i \parallel j.$$

   Equivalently, no strictly higher-priority constraint is overridden by a strictly lower-priority comparable constraint.

2. ***Persistence of maximal constraints under non-interference.*** *Under* (NI)*, every maximal constraint of the poset is satisfied by the final output $\boldsymbol{u}_{\mathrm{out}}$.*

3. ***Convex mixing of maximal-safety heads.*** *Consider K heads, each using a linear extension of $\preceq$, and let $\boldsymbol{u}_r^{\mathrm{out}}$ be the output of head r. If*

   $$\bar{\boldsymbol{u}} := \sum_{r=1}^{K} \theta_r \boldsymbol{u}_r^{\mathrm{out}}, \qquad \theta_r \geq 0, \qquad \sum_{r=1}^{K} \theta_r = 1,$$

   then $\bar{\boldsymbol{u}}$ satisfies all maximal constraints of the poset.

Proof. We prove the three claims separately.

1. Poset-respecting override. Fix a constraint i, and suppose that i has already been enforced at step s, so $j_s = i$ and

$$\boldsymbol{u}^{(s)} \in H_i.$$

If i becomes violated at a later step $t > s$, then the violating projection is onto constraint j_t:

$$\boldsymbol{u}^{(t)} = \Pi_{j_t}\big(\boldsymbol{u}^{(t-1)}\big), \qquad \boldsymbol{u}^{(t-1)} \in H_i, \qquad \boldsymbol{u}^{(t)} \notin H_i.$$

Because j_t appears after i in the linear extension σ, it cannot be the case that $j_t \prec i$; otherwise, the definition of a linear extension would require j_t to appear before i. Hence either $i \prec j_t$ or $i \parallel j_t$.

Table 5: Full trajectory-level non-interference audit.

Task	Implementation	Interference events / eligible events	Worst margin
Navigation	PoSafeNet (Hard)	$0/103{,}488$	-6.06×10^{-9}
Navigation	PoSafeNet (Mixture)	$0/640{,}512$	-1.94×10^{-9}
Manipulation	PoSafeNet (Hard)	$0/35{,}848$	-4.86×10^{-9}
Manipulation	PoSafeNet (Mixture)	$0/71{,}696$	-9.14×10^{-9}
Driving	PoSafeNet (Hard)	$0/300$	-2.56×10^{-8}
Driving	PoSafeNet (Mixture)	$0/1{,}200$	-6.46×10^{-8}

Therefore, a constraint that has already been enforced can only be overridden by a later higher-priority constraint or by an incomparable constraint selected later in the admissible execution. In particular, no strictly higher-priority constraint is overridden by a strictly lower-priority comparable constraint.

2. Persistence of maximal constraints. Let i be maximal in $(\mathcal{S}, \preceq)$, and let s be its position in σ. Since step s projects onto H_i, we have

$$\boldsymbol{u}^{(s)} \in H_i.$$

Consider any later step $t > s$, with projection onto j_t. Since i is maximal, there is no constraint j_t such that $i \prec j_t$. Also, because j_t appears after i, we cannot have $j_t \prec i$; otherwise, the linear-extension property would require j_t to appear before i. Therefore $j_t \parallel i$. By the non-interference condition (NI), projection onto H_{j_t} preserves membership in H_i. Inducting over all later steps yields

$$\boldsymbol{u}_{\text{out}} \in H_i.$$

Since i was arbitrary, all maximal constraints are satisfied by the final output.

3. Convex mixing of maximal-safety heads. By Part 2, for every maximal constraint i and every head r,

$$\boldsymbol{u}_r^{\text{out}} \in H_i.$$

Because each H_i is convex, any convex combination of points in H_i remains in H_i. Therefore,

$$\bar{\boldsymbol{u}} = \sum_{r=1}^{K} \theta_r \boldsymbol{u}_r^{\text{out}} \in H_i$$

for every maximal constraint i. Hence the mixed output preserves all maximal, i.e., highest-priority, safety constraints. □

C.5 Trajectory-Level Non-Interference Audit

The navigation obstacle constraints are mutually incomparable and co-maximal; the manipulation joint-limit constraints are likewise incomparable and co-maximal. We audited all directed preservation implications in every evaluated head using float64. For a directed incomparable pair (i, j), define the margin $m_j(u) = A_j u - c_j$. An event is eligible when $m_j(u) \geq -\tau$ before projection Π_i, and interference is recorded when $m_j(\Pi_i(u)) < -\tau$. In four-layer driving, obstacle avoidance is projected last, so its guarantee does not rely on non-interference; the driving results are included for completeness.

No interference events were observed at the audit tolerances on the evaluated trajectories.

Antiparallel bound pairs. The normal dot-product sufficient condition does not apply to the lane, speed, centerline, and joint-limit bound pairs because their normals are antiparallel. For exact Euclidean projections with nonzero antiparallel normals, bidirectional non-interference is equivalent to $H_i \cap H_j \neq \emptyset$. We therefore audited intersection feasibility.

The maximum antiparallel-normal error was 0 for driving and 2.22×10^{-16} for manipulation. No evaluated pair had an empty intersection.

Numerical precision and tolerance. All audits used float64 and tolerances above the measured projection residuals. Table 7 reports the task-specific tolerances and maximum residuals for Hard and Mixture.

Table 6: Intersection-feasibility audit for antiparallel bound pairs.

Task	Implementation	Bound pair	Empty intersections / checks	Minimum intersection slack
Driving	PoSafeNet (Hard)	Centerline, lane, and speed bounds	0/300	0.1376
Driving	PoSafeNet (Mixture)	Centerline, lane, and speed bounds	0/1,200	0.1409
Manipulation	PoSafeNet (Hard)	Joint-limit bounds	0/35,848	13.0325
Manipulation	PoSafeNet (Mixture)	Joint-limit bounds	0/71,696	13.0325

Table 7: Numerical tolerances and maximum projection residuals used in the float64 audits.

Task	Tolerance	Maximum residual (Hard / Mixture)
Navigation	10^{-7}	6.63×10^{-10} / 1.47×10^{-9}
Manipulation	10^{-7}	3.17×10^{-9} / 7.23×10^{-9}
Driving	10^{-6}	2.56×10^{-8} / 6.46×10^{-8}

C.6 Limitations and the Role of Posets

The above conditions are sufficient but not universal. When multiple hazards induce opposing control corrections, for example constraints requiring steering in opposite directions or competing accelerations, Condition C1 fails and the control-space normals may satisfy

$$A_i A_j^\top < 0.$$

In this case, enforcing one CBF can degrade or violate another.

Such intrinsically conflicting constraints should therefore be ordered or aggregated rather than treated as a mutually compatible antichain. The poset structure in our method reflects this geometric compatibility between safety constraints, rather than being an arbitrary design choice.

D Experimental Details

Task-specific supervision. Training uses control imitation without CBF-value or constraint-label supervision. Table 8 summarizes the expert controls, the constraints enforced by each expert, the constraints added only by PoSafeNet, and the state information used to construct the CBFs.

Evaluation metrics. Unless otherwise specified, quantitative results are computed over $N = 100$ closed-loop evaluation rollouts. For each rollout k, let $\boldsymbol{x}^{(k)}(t)$ denote its state trajectory, and let $b(\boldsymbol{x}) \geq 0$ define the task-specific safety constraint.

Safety (min / mean). We report the minimum and mean safety margin over time and across rollouts, defined as

$$\text{Safety}_{\min} = \min_k \left\{ \min_{t \in [t_0, T]} b(\boldsymbol{x}^{(k)}(t)) \right\}, \qquad \text{Safety}_{\text{mean}} = \mathbb{E}_k \left[\mathbb{E}_{t \in [t_0, T]} \left[b(\boldsymbol{x}^{(k)}(t)) \right] \right].$$

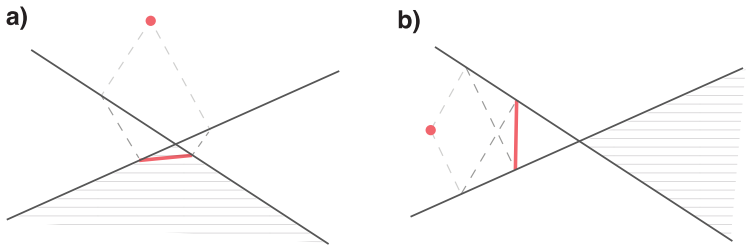


Figure 5: **Geometric intuition for compatibility and conflict between CBF constraints.** (**a**) Two CBF halfspaces with non-opposing normals ($A_i A_j^\top \geq 0$). Sequential projection onto one constraint does not violate the other, and the resulting control lies in the intersection of the halfspaces; convex mixing of such controls therefore preserves feasibility. (**b**) Conflicting CBF constraints with opposing normals ($A_i A_j^\top < 0$), where enforcing one constraint may degrade or violate the other. Such constraints should not be treated as a mutually compatible antichain and may need to be ordered or prioritized, motivating poset-structured safety enforcement.

Table 8: Task-specific supervision and state information used to construct the CBFs.

Task	Expert control	Expert constraints	Constraints added only by PoSafeNet	CBF state
Navigation	HOCBF-QP $[\dot{\theta}, \dot{v}]$	Obstacle avoidance	None	$[p_x, p_y, \theta, v]$ and obstacle geometry
Manipulation	HOCBF-QP $[\ddot{\theta}_1, \ddot{\theta}_2]$	End-effector obstacle avoidance	Relative joint limits	$[\theta_1, \omega_1, \theta_2, \omega_2]$, obstacle geometry
Driving	NMPC controls paired with RGB	Obstacle avoidance	Lane, speed, and centerline constraints	Simulator state $[s, d, \mu, v, \delta]$, lane geometry, and obstacle positions

Negative values indicate violation of the corresponding safety constraint.

Top Safety. This empirical metric indicates whether the evaluated top-priority safety constraint is satisfied at every evaluated time step across all rollouts. For obstacle avoidance, this corresponds to a nonnegative minimum obstacle-barrier margin. For joint-space safety, it indicates zero ϕ-violation across the evaluated rollouts. This metric reports observed constraint satisfaction, rather than a formal forward-invariance guarantee. **Uncertainty.** Control uncertainty is quantified as the time-averaged standard deviation of control inputs across rollouts,

$$\text{Unc.}(\boldsymbol{u}_i) = \mathbb{E}_t \left[\text{std}_k \left(\boldsymbol{u}_i^{(k)}(t) \right) \right], \quad i \in \{1, 2\}.$$

Feasibility. The *Feasibility* metric indicates whether a valid control input is successfully produced at *every* time step along a closed-loop rollout. A rollout is deemed infeasible if the underlying safety layer fails to return a control input at any time step, for example due to optimization infeasibility, numerical instability, or solver failure. This metric therefore captures the operational reliability of the safety enforcement mechanism, independent of the achieved safety margin.

D.1 2D Robot Multi-Obstacle Avoidance

To evaluate robustness to observation noise, we add zero-mean Gaussian noise to the normalized network input. Both the nominal control and learned CBF gains depend on the noisy observation, while the state and geometry used to construct the CBF constraints remain unperturbed simulator ground truth. For each of the 24 held-out test episodes, we perform multiple stochastic rollouts, resulting in a total of $N = 100$ evaluation runs.

Models. All models use fully connected networks with architecture $[5, 128, 32, 32, 2]$ and `ReLU` activations. There are some additional layers of differentiable QPs in other models (other than E2E-related models). The model input consists of the system state and the goal.

Training and Dataset. For the 2D robot multi-obstacle avoidance task, the training set consists of 184 trajectories (66,420 state–control pairs), while the held-out test set contains 24 trajectories (8,790 samples), each with approximately 360 time steps. Ground-truth controls are generated by solving higher-order control barrier function (HOCBF)-based quadratic programs that enforce all three obstacle-avoidance constraints simultaneously. All models are trained using the `Adam` optimizer with an MSE loss and a learning rate of 10^{-3}. Differentiable QP layers are implemented using `qpth` from OptNet. The training time for both PoSafeNet variants (mixture and hard) is approximately one minute for 20 epochs. For multi-head models, ABNet uses 5 heads, while PoSafeNet uses 6 heads.

Robot dynamics and safety constraints. We employ the unicycle model as the robot dynamics:

$$\underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{v}(t) \end{bmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}}_{\boldsymbol{u}}, \tag{18}$$

where $(x, y) \in \mathbb{R}^2$ denotes the planar position of the robot, $\theta \in \mathbb{R}$ is the heading angle, $v \in \mathbb{R}$ is the linear speed, and u_1 and u_2 are the angular velocity and acceleration controls, respectively.

The obstacle-avoidance safety constraint is defined as

$$b(\boldsymbol{x}) = (x - x_0)^2 + (y - y_0)^2 - R^2 \geq 0, \tag{19}$$

where $(x_0, y_0) \in \mathbb{R}^2$ denotes the obstacle center and $R > 0$ is its radius.

D.2 Safe Robot Manipulation under Joint Constraints

To evaluate robustness to observation noise, we add zero-mean Gaussian noise to the normalized network input. Both the nominal control and learned CBF gains depend on the noisy observation, while the state and geometry used to construct the CBF constraints remain unperturbed simulator ground truth. For each of the 20 held-out test episodes, we perform multiple stochastic rollouts, resulting in a total of $N = 100$ evaluation runs.

Models. All models use fully connected neural networks with architecture $[6, 128, 256, 128, 32, 32, 2]$ and `ReLU` activation functions. Unstructured safety baselines additionally include differentiable QP layers, whereas the end-to-end (E2E) baseline does not enforce explicit safety constraints.

Training and Dataset. For the safe robot manipulation task, the training set consists of 160 trajectories with approximately 380 time steps each (60,614 state–control pairs). The held-out test set contains 20 trajectories with approximately 370 time steps per trajectory (7,329 samples). Ground-truth controls are generated by solving higher-order control barrier function (HOCBF)-based quadratic programs that enforce only end-effector (obstacle-avoidance) safety, without joint-angle constraints. All models are trained using the `Adam` optimizer with an MSE loss and a learning rate of 10^{-3}. Differentiable QP layers are implemented using `qpth` from OptNet. The training time for both PoSafeNet variants (mixture and hard) is approximately 24 seconds for 20 epochs. For multi-head models, ABNet uses 5 heads, while PoSafeNet uses 2 heads.

Robot dynamics and safety constraints. We model the two-link planar manipulator using the following control-affine dynamics:

$$\underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\omega}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_2 \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \omega_1 \\ 0 \\ \omega_2 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}}, \tag{20}$$

where $(\theta_1, \theta_2) \in \mathbb{R}^2$ denote the joint angles, $(\omega_1, \omega_2) \in \mathbb{R}^2$ are the joint angular velocities, and u_1, u_2 are the joint angular acceleration controls.

The end-effector obstacle-avoidance safety constraint is defined as

$$b_{\text{tip}}(\boldsymbol{x}) = (l_1 \cos\theta_1 + l_2 \cos\theta_2 - x_0)^2 + (l_1 \sin\theta_1 + l_2 \sin\theta_2 - y_0)^2 - R^2 \geq 0, \tag{21}$$

where $(x_0, y_0) \in \mathbb{R}^2$ denotes the obstacle center, $R > 0$ is its radius, and $l_1, l_2 > 0$ are the link lengths. In the current setting, ensuring end-effector safety also guarantees non-collision of the links.

In addition to end-effector safety, we impose joint-angle constraints to prevent excessive bending of the manipulator. We define the relative joint angle

$$\phi = \text{wrap}(\theta_2 - \theta_1) \in [-\pi, \pi), \tag{22}$$

where $\text{wrap}(\cdot)$ maps angles to the principal interval $[-\pi, \pi)$. The corresponding angular velocity is $\dot{\phi} = \omega_2 - \omega_1$.

We enforce joint-angle limits by constraining ϕ to lie within a prescribed interval $[\phi_{\min}, \phi_{\max}]$ using two higher-order control barrier functions:

$$b_\phi^{\min}(\boldsymbol{x}) = \phi - \phi_{\min} \geq 0, \tag{23}$$

$$b_\phi^{\max}(\boldsymbol{x}) = \phi_{\max} - \phi \geq 0. \tag{24}$$

Since ϕ has relative degree two with respect to the control input, with $\ddot{\phi} = u_2 - u_1$, each constraint is enforced via a second-order control barrier function (HOCBF).

These joint-angle constraints are treated as higher-priority safety requirements than the end-effector obstacle-avoidance constraint in the poset, reflecting the fact that joint-limit violations are unacceptable even when obstacle avoidance could otherwise be satisfied.

D.3 Vision-Based End-to-End Autonomous Driving

Models. All methods share the same perception backbone, consisting of a convolutional neural network followed by an LSTM and fully connected control heads. The CNN comprises five convolutional layers with configurations $[3, 24, 5, 2, 2]$, $[24, 36, 5, 2, 2]$, $[36, 48, 3, 2, 1]$, $[48, 64, 3, 1, 1]$, and $[64, 64, 3, 1, 1]$, where each tuple specifies the number of input channels, output channels, kernel size, stride, and padding, respectively. The convolutional features are processed by an LSTM with hidden size 64 to capture temporal dependencies. The control head consists of two fully connected layers of size $[32, 32, 2]$ with `ReLU` activations, producing the steering rate and acceleration commands.

Dropout with rate 0.3 is applied to both convolutional and fully connected layers during training. Unstructured safety baselines additionally include differentiable QP layers, whereas the end-to-end (E2E) baseline does not incorporate explicit safety constraints. The model input is the front-view RGB image from the ego vehicle with resolution $3 \times 45 \times 155$, and the output is a two-dimensional control vector corresponding to steering rate and longitudinal acceleration.

Training and Dataset. We use a publicly available dataset containing approximately 0.4 million image–control pairs collected in a closed-road sim-to-real driving environment. The dataset is generated using the `VISTA` simulator [3], and includes static and parked vehicles of varying types and appearances as obstacles. Ground-truth control commands are obtained by solving a nonlinear model predictive control (NMPC) problem, which serves as the expert policy for imitation learning. All models are trained using the `Adam` optimizer with an MSE loss and a learning rate of 10^{-3}. For methods incorporating differentiable safety layers, quadratic programs are solved using `qpth` from OptNet [4]. Training the PoSafeNet requires approximately 50 minutes for 10 epochs on a single NVIDIA RTX 5090 GPU.

Brief introduction to `VISTA`. `VISTA` is a sim-to-real driving simulator that generates diverse driving scenarios from real-world driving data [3]. We train vision-based policies via guided imitation learning using an expert controller. Data are generated in three steps: (i) `VISTA` randomly initializes the poses of the ego vehicle and other traffic participants (ado vehicles) consistent with the underlying real driving logs; (ii) a nonlinear model predictive controller (NMPC) is used to compute expert controls and the corresponding state trajectories; and (iii) front-view RGB images are recorded along NMPC rollouts and paired with the expert control commands.

Vehicle dynamics and collision safety constraints. Vehicle dynamics are formulated in a curvilinear coordinate frame defined with respect to a reference trajectory (e.g., the lane centerline) [14]. The system state is $x = [s, d, \mu, v, \delta]^\top$, where $s \in \mathbb{R}$ denotes the progress along the reference trajectory, $d \in \mathbb{R}$ is the lateral offset of the vehicle center, μ is the heading error with respect to the reference trajectory, v is the longitudinal speed, and δ is the steering angle. The control input is $u = [u_1, u_2]^\top$, where u_1 is the steering rate and u_2 is the longitudinal acceleration.

We adopt the curvilinear bicycle model in [14], which can be written in control-affine form as

$$\underbrace{\begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\mu} \\ \dot{v} \\ \dot{\delta} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \dfrac{v\cos(\mu+\beta)}{1-d\kappa(s)} \\ v\sin(\mu+\beta) \\ \dfrac{v}{\ell_r}\sin\beta - \dfrac{\kappa(s)\,v\cos(\mu+\beta)}{1-d\kappa(s)} \\ 0 \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}}. \tag{25}$$

Collision avoidance is encoded via a control barrier function defined in the curvilinear frame:

$$b_{\mathrm{col}}(\boldsymbol{x}) = (s - s_0)^2 + (d - d_0)^2 - R^2 \geq 0, \tag{26}$$

where $(s_0, d_0) \in \mathbb{R}^2$ denotes the obstacle location expressed in curvilinear coordinates and $R > 0$ is a safety radius chosen such that satisfying (26) guarantees collision avoidance.

Lane-keeping constraints. Lane keeping is enforced by constraining the lateral offset d of the ego vehicle with respect to the reference trajectory. Specifically, we require the vehicle to remain within a lateral margin $\ell_f > 0$ around the lane centerline:

$$-\ell_f \le d \le \ell_f. \tag{27}$$

This is implemented using two higher-order control barrier functions corresponding to the left and right lane boundaries. For the left boundary, we define

$$b_{\mathrm{L}}(\boldsymbol{x}) = \ell_f - d \ge 0, \tag{28}$$

and for the right boundary,

$$b_{\mathrm{R}}(\boldsymbol{x}) = d + \ell_f \ge 0. \tag{29}$$

Both constraints are enforced using second-order barrier conditions that account for the vehicle dynamics in the curvilinear coordinate frame, resulting in affine constraints on the control input at each time step.

Centerline-adherence constraints. In addition to lane-boundary safety, we include a lower-priority centerline-adherence objective that encourages the ego vehicle to remain close to the lane center. This objective is encoded as a pair of soft safety-style constraints on the lateral offset d:

$$-\ell_c \le d \le \ell_c, \qquad 0 < \ell_c < \ell_f, \tag{30}$$

where ℓ_c defines a narrow tolerance band around the centerline.

Similar to the lane-keeping constraints, we implement centerline adherence using two higher-order barrier conditions. The left centerline constraint is

$$b_{\mathrm{CL}}(\boldsymbol{x}) = \ell_c - d \ge 0, \tag{31}$$

and the right centerline constraint is

$$b_{\mathrm{CR}}(\boldsymbol{x}) = d + \ell_c \ge 0. \tag{32}$$

These constraints induce affine inequalities in the control input under the curvilinear vehicle dynamics. Unlike the lane-boundary constraints, however, centerline adherence is treated as a lower-priority objective in the safety poset: it may be overridden to preserve lane keeping, speed regulation, or obstacle avoidance when conflicts occur.

Closed-loop testing. All methods are evaluated closed-loop in `VISTA`. At each time step, the policy receives a front-view RGB observation and outputs a control command, which is applied to the simulator. Rollouts terminate when the horizon is reached (or upon collision, when applicable). Unless otherwise stated, we report results over $N = 100$ closed-loop runs.

For each run, the ego vehicle is initialized on the straight segment with $s_0 \sim \mathcal{U}(0, 10)\,\mathrm{m}$ and lateral offset $d \sim \mathcal{U}(-0.5, 0.5)\,\mathrm{m}$. Two obstacles are placed ahead of the ego vehicle with longitudinal offsets $\Delta s_1 \sim \mathcal{U}(20, 30)\,\mathrm{m}$ and $\Delta s_2 = \Delta s_1 + \mathcal{U}(30, 40)\,\mathrm{m}$, and lateral offsets sampled uniformly from $\{|d_0| \sim \mathcal{U}(0.1, 1.5)\}\,\mathrm{m}$ with random sign.

HQP baseline. The baseline uses a lexicographic HQP with level-specific relaxation. Later levels fix higher-priority relaxation vectors componentwise. The four levels are: (1) obstacle avoidance, (2) lane bounds, (3) speed bounds, and (4) centerline bounds.

Let $G_\ell u \le h_\ell$ denote the constraints at level ℓ. The obstacle level solves

$$u_1^\star = \arg\min_u \|u\|^2 \quad \text{s.t.} \quad G_1 u \le h_1.$$

For $\ell > 1$, we solve

$$\begin{aligned}(u_\ell^\star, \delta_\ell^\star) = \ & \arg\min_{u,\, \delta_\ell \ge 0} \|\delta_\ell\|^2 + \epsilon\|u\|^2 \\ \text{s.t.} \quad & G_1 u \le h_1, \\ & G_j u \le h_j + \delta_j^\star, \qquad j = 2, \dots, \ell - 1, \\ & G_\ell u \le h_\ell + \delta_\ell,\end{aligned}$$

Table 9: Per-constraint violation rates at matched HQP progress over 10 paired blocking-obstacle scenarios.

Method	Obstacle vio.	Lane vio.	Speed vio.	Progress (m)
HQP	0%	0%	0%	27.14 ± 4.39
PoSafeNet (Mixture)	0%	0%	1.78%	27.48 ± 4.39
PoSafeNet (Hard)	0%	0%	13.98%	27.49 ± 4.46

where $\epsilon > 0$ is a small regularization parameter. Each component of δ_ℓ relaxes one inequality. Later levels preserve the previously fixed higher-priority relaxation values.

The final control solves

$$\begin{aligned} u^\star = \; & \arg\min_u \|u - u_{\text{nom}}\|^2 \\ \text{s.t.} \quad & G_1 u \le h_1, \\ & G_j u \le h_j + \delta_j^\star, \qquad j = 2, \dots, L, \end{aligned}$$

with $L = 4$.

In blocking-obstacle scenarios, stopping can satisfy the obstacle constraint and attain zero lane relaxation. Once these higher-priority relaxation values are fixed, lower-priority objectives cannot increase lane relaxation to pass the obstacle, explaining possible stalling.

Neither HQP nor PoSafeNet uses post-optimization clipping or actuator-limit constraints in these experiments. The nominal policy has the same architecture as in PoSafeNet and enters HQP only through the final tracking objective. In contrast, the nominal control is the starting point for each projection head in PoSafeNet. Thus, the progress gap reflects both the constraint-resolution mechanism and where the nominal objective enters, rather than the poset structure alone.

We use the `qpth` float64 batched primal-dual solver with `eps`$= 10^{-12}$, `maxIter`$= 20$, and `notImprovedLim`$= 3$.

Per-constraint outcomes at matched progress. For 10 paired blocking-obstacle scenarios, we use the maximum progress reached by HQP as the episode-specific target. Each method's trajectory is truncated at the first timestep reaching that target. Table 9 reports timestep-level violation rates over these matched trajectory prefixes. The small differences in reported progress arise from discrete-time truncation at the first timestep reaching the target.

At matched HQP progress, all three methods exhibit zero obstacle and lane violations. HQP also preserves the speed constraints, whereas Mixture and Hard exhibit speed-violation rates of 1.78% and 13.98%, respectively. All observed Hard speed violations are upper-speed violations. These results do not establish uniform dominance over HQP; they characterize the constraint trade-off at this matched-progress operating point.

E Additional Ablations on Projection-Order Sensitivity and Control Quality (VISTA)

Effect of Cross-Priority Ordering. To study the sensitivity of sequential safety projection to constraint ordering, we compare four variants: a fixed admissible projection order, an intentionally reversed cross-priority order, learned hard selection over admissible heads, and mixture-based combination of admissible heads.

In **PoSafeNet (Fixed Poset Order)**, the left and right lane-boundary constraints are projected first in a fixed, manually specified order, followed by obstacle avoidance. Since constraints projected later have higher effective priority, projecting obstacle avoidance last realizes the intended hierarchy in which collision avoidance strictly dominates lane keeping.

In **PoSafeNet (Cross-Priority Swap)**, obstacle avoidance is instead projected before the lane-boundary constraints. The later lane projections may therefore modify or overwrite the obstacle-avoidance correction. This variant intentionally violates the specified cross-priority relation and isolates the effect of placing lower-priority constraints after a higher-priority constraint.

Table 10: **Ablation results for VISTA vision-based autonomous driving.** Results are reported for a single training seed. All variants in this table use the same evaluation protocol (Appendix D). The scenario set and evaluation protocol differ from those used in Table 4; numerical results should therefore not be directly compared across the two tables. Obstacle safety is quantified by the minimum and mean signed obstacle-barrier margin over each rollout, where higher values are better and negative values indicate violation. Lane keeping is quantified by the minimum and mean signed lane margin, where higher values are better and negative values indicate lane departure. We additionally report the mean and variance of the absolute lateral deviation $|e_y|$ from the lane centerline, together with the fraction of time spent outside the lane. *Pass%* denotes the percentage of rollouts completed without collision, while *Crash%* denotes the percentage terminating in collision. *Rot. magnitude* is the mean rotation-rate magnitude, and *Uncertainty* (u_1, u_2) denotes the temporal standard deviation of the two control inputs. Higher is better for Pass, obstacle margin, and lane margin; lower is better for Crash, $|e_y|$, time-out-of-lane, rotational magnitude, and uncertainty.

Method	Pass% ↑	Crash% ↓	Obstacle margin (min / mean) ↑	Lane margin (min / mean) ↑	$\lvert e_y \rvert$ (mean / var) ↓	Time-out-of-lane ↓	Rot. magnitude ↓	Unc. (u_1, u_2) ↓
PoSafeNet (Fixed Poset Order)	100.0	0.0	0.27 / 3.38	−6.15 / −1.71	1.73 / 3.09	0.3197	0.65	(0.412, 0.641)
PoSafeNet (Cross-Priority Swap)	41.0	59.0	−24.42 / −17.70	0.06 / 0.14	0.989 / 0.746	0.0000	0.00	(0.108, 0.742)
PoSafeNet (Mixture)	100.0	0.0	1.23 / 8.21	−0.93 / −0.49	1.028 / 1.278	0.1992	0.14	(0.287, 0.387)
PoSafeNet (Hard)	100.0	0.0	2.73 / 7.12	−0.63 / −0.32	1.404 / 1.599	0.3670	0.80	(0.345, 0.456)

Although both fixed-order variants use the same safety constraints, reversing their cross-priority projection order leads to qualitatively different outcomes (Table 10). The cross-priority-swap variant maintains positive lane margins and remains within the lane, but exhibits severe degradation in obstacle-safety margins and collision performance. This shows that a lower-priority constraint projected later can override a previously enforced higher-priority constraint, even when the individual constraint projections are well defined.

In contrast, **PoSafeNet (Fixed Poset Order)** remains collision-free with positive obstacle-safety margins, but incurs substantial lane departures. Because obstacle avoidance is projected last, its correction may override the earlier lane-keeping corrections when the constraints conflict. This behavior illustrates the limitation of using one rigid, manually selected admissible linear extension.

PoSafeNet (Hard) learns global logits over multiple admissible projection heads using a Gumbel-Hard relaxation during training. At inference, the model deterministically selects one fixed global linear extension for all states; the selected ordering is not state-dependent. In this single-seed evaluation, Hard achieves higher obstacle-safety and lane margins than the fixed admissible order, although its fraction of time spent outside the lane is higher. In a separate within-antichain order-swap ablation, the lateral bias did not reverse when the left and right lane projection orders were swapped. Thus, the observed bias cannot be attributed solely to which lane constraint is projected last.

Finally, **PoSafeNet (Mixture)** combines the outputs of multiple admissible heads using learned convex weights. In this single-seed evaluation, Mixture achieves lower mean absolute lateral deviation, rotational magnitude, and time-out-of-lane than Hard, while both remain collision-free. These are empirical differences; the cause of the observed Hard–Mixture difference remains unresolved.

E.1 Ablation on the Number of Heads

We evaluate PoSafeNet (Mixture) with $H \in \{1, 2, 4, 8\}$ on the four-layer driving task, keeping the remaining training configuration and evaluation setting unchanged. This is a single-seed ablation, with each model evaluated on the same 10 benchmark episodes. All heads preserve the cross-layer projection order

$$\text{centerline} \;\rightarrow\; \text{speed} \;\rightarrow\; \text{lane} \;\rightarrow\; \text{obstacle},$$

with the highest-priority obstacle-avoidance constraint projected last. The heads differ only in the ordering of the two incomparable lane constraints and the two incomparable speed constraints.

For $H = 1$, we use $v_{\max} \rightarrow v_{\min}$ and $L \rightarrow R$. For $H = 2$, we additionally include $v_{\max} \rightarrow v_{\min}$ and $R \rightarrow L$. For $H = 4$, we include all four combinations of $L \rightarrow R$ or $R \rightarrow L$ and $v_{\max} \rightarrow v_{\min}$ or $v_{\min} \rightarrow v_{\max}$. For $H = 8$, we assign two heads to each of these four unique orders.

All settings maintain positive obstacle-safety margins and zero crashes in this evaluation. The single-head model spends 10.2% of its execution time outside the lane and achieves progress of 49.93. Increasing to $H = 2$ eliminates the observed lane departures and increases progress to 61.72. The $H = 2$ and $H = 4$ settings achieve similar lane tracking and progress, while duplicating the

Table 11: Head-count ablation for PoSafeNet (Mixture) on the four-layer VISTA driving task. Results use one training seed and the same 10 benchmark episodes for each setting.

H	Obstacle Safety (min) ↑	Crash ↓	Lane margin (min) ↑	Time ratio out-of-lane ↓	Progress ↑	$\lvert e_y \rvert$ (mean) ↓
1	2.95	0.0	−0.09	0.102	49.93	0.96
2	1.38	0.0	0.31	0.000	61.72	0.47
4	1.84	0.0	0.29	0.000	61.64	0.47
8	1.53	0.0	0.29	0.000	57.81	0.55

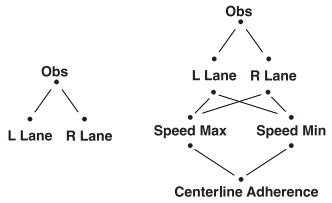


Figure 6: **Task-specific safety posets used in the vision-based driving experiments, visualized as Hasse diagrams.** Left: a two-level hierarchy where obstacle avoidance has higher priority than left and right lane boundary constraints. Right: a four-level hierarchy that augments the two-level structure with intermediate speed regulation and low-priority centerline adherence, while preserving obstacle avoidance as the highest-priority constraint.

four orders at $H = 8$ provides no further benefit in this run. We use $H = 4$ to represent all four distinct extensions under the fixed cross-layer priority structure. These observations are limited to this single-seed ablation; a cross-seed performance ordering is not established.

F Four-Level Poset Safety Hierarchy for Heatmap Visualization

This appendix describes the four-level safety hierarchy used to generate the heatmap visualizations in the vision-based driving experiments. The purpose is not to introduce new modeling assumptions or additional safety constraints, but to illustrate how refining the priority structure within the same poset-based framework influences closed-loop behavior. In particular, this example highlights the ability of PoSafeNet to incorporate additional low-priority preferences without compromising higher-priority safety requirements.

F.1 Poset Structure

Figure 6 compares the task-specific safety posets used in the experiments. The two-level hierarchy enforces only road boundary constraints and obstacle avoidance, whereas the four-level hierarchy further decomposes driving behavior into multiple priority tiers that encode softer preferences.

Formally, we use the partial order symbol $\preceq$ to denote *priority relations* between constraints, where $c_1 \preceq c_2$ indicates that constraint c_1 has lower or equal priority than constraint c_2. Under this convention, the four-level hierarchy orders constraints from lowest to highest priority as

$$\text{Centerline} \preceq \text{Speed} \preceq \text{Lane} \preceq \text{Obstacle}. \tag{33}$$

Low-priority constraints, such as centerline adherence and speed regulation, are treated as soft behavioral preferences. They are enforced only when doing so does not interfere with higher-priority safety constraints, such as lane boundaries and obstacle avoidance.

F.2 Additional Barrier Definitions

In addition to the lane boundary and obstacle avoidance barriers defined elsewhere, the four-level hierarchy introduces two *additional low-priority* constraints to better capture common driving preferences.

Centerline adherence constraints. Centerline adherence is modeled as a soft preference that encourages the vehicle to remain close to the lane centerline. Let d denote the signed lateral deviation

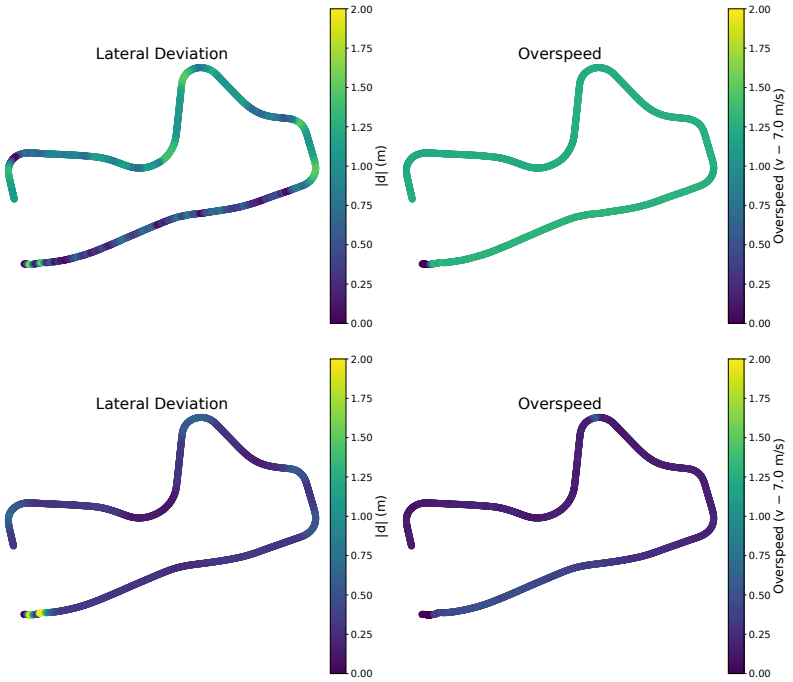


Figure 7: **Heatmap visualization of closed-loop driving behavior under different safety hierarchies.** Top: two-level poset (lane $\preceq$ obstacle). Bottom: four-level poset (centerline $\preceq$ speed $\preceq$ lane $\preceq$ obstacle). Colors indicate lateral deviation and overspeed along the trajectory.

from the lane centerline and $\tau > 0$ a prescribed tolerance. We define two symmetric barrier functions

$$b_{\mathrm{CL},L}(\boldsymbol{x}) = \tau - d \geq 0, \qquad b_{\mathrm{CL},R}(\boldsymbol{x}) = d + \tau \geq 0. \tag{34}$$

These barriers share the same higher-order control barrier function (HOCBF) structure as the lane boundary constraints, but are enforced at a strictly lower priority level.

Speed regulation constraints. Speed regulation is introduced as a low-priority constraint to discourage overspeeding and excessive braking. Let v denote the vehicle speed and $v_{\min}, v_{\max}$ the desired bounds. We define first-order control barrier functions

$$b_{\max}(v) = v_{\max} - v \geq 0, \qquad b_{\min}(v) = v - v_{\min} \geq 0. \tag{35}$$

These constraints yield affine conditions on the longitudinal acceleration and are enforced at a lower priority than lane boundary and obstacle avoidance constraints.

F.3 Interpretation of Heatmaps

The heatmaps illustrate how introducing additional low-priority structure alters the distribution of closed-loop behaviors. Compared to the two-level hierarchy, the four-level hierarchy exhibits reduced lateral deviation and lower overall speed, while preserving collision avoidance and lane safety.

These results demonstrate that PoSafeNet can absorb additional behavioral preferences by refining the poset hierarchy, shaping closed-loop behavior in a more structured and human-like manner without compromising higher-priority safety constraints. This example therefore serves as a qualitative illustration of the extensibility of the poset-based formulation.

G Supplementary On Vision Based End-to-End Autonomous Driving

This section provides qualitative visualizations of closed-loop driving behaviors in the VISTA simulator for representative methods. Each snapshot shows the front-view RGB observation received

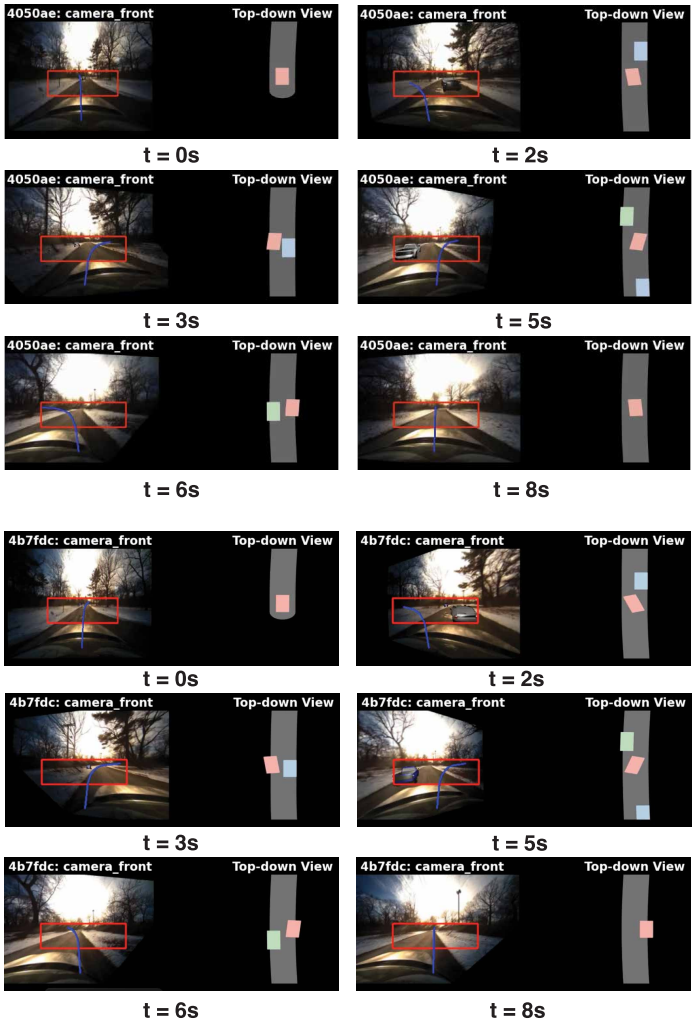


Figure 8: **Qualitative visualization of closed-loop driving behaviors in the VISTA simulator.** Each snapshot corresponds to a specific time step along the trajectory (timestamps shown above each panel). For each time step, the left image shows the front-view camera observation, while the right image shows the corresponding top-down view. From top to bottom: PoSafeNet (Mixture) and PoSafeNet (Hard).

by the policy (left) and a corresponding top-down view of the scene (right), including the ego vehicle, lane centerline, and surrounding obstacles.

These visualizations are intended to complement the quantitative results reported in the main paper by illustrating typical driving behaviors under dynamic obstacle interactions, rather than to serve as a standalone performance comparison.

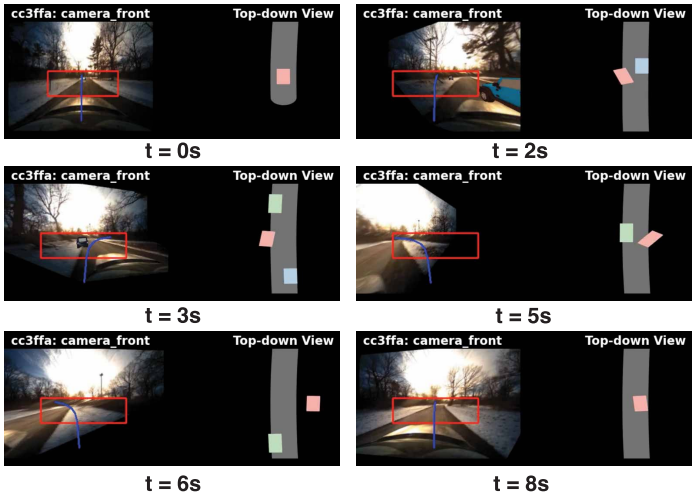


Figure 9: **Qualitative visualization of closed-loop driving behaviors in the VISTA simulator using ABNet.** Each snapshot corresponds to the same set of time steps as in Fig. 8 (timestamps shown above each panel). The left image shows the front-view camera observation and the right image shows the corresponding top-down view. Compared with PoSafeNet, ABNet without lane keeping constraint exhibits less consistent lane tracking.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims and contributions made in the abstract and introduction are all supported by theoretical and/or experimental results included in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a "Limitations and Future Work" discussion that describes the current scope of PoSafeNet, including the use of task-level safety posets, CBF-induced control halfspaces, output-space safety composition, and future extensions to learned posets, perception-derived constraints, stochastic dynamics, multi-agent interaction, and contact-rich manipulation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main text states the assumptions for poset-respecting safety, linear extensions, hard head selection, and convex mixture safety. Proofs for the main propositions and theorem are provided in the main text, and additional geometric conditions for sequential projection and mixture safety are provided in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the evaluated tasks, baselines, safety constraints, model architectures, datasets, training procedures, rollout settings, metrics, and compute setup. Additional implementation and experimental details are provided in the appendix, including dynamics, barrier functions, network architectures, dataset sizes, noise settings, optimizer choices, and evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides detailed algorithmic, architectural, and experimental information needed to reproduce the main results, and the main code is released with the initial anonymous submission. The VISTA driving simulator and associated dataset are publicly available and cited; other experimental details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper and appendix specify the training and evaluation settings for all experiments, including task dynamics, safety constraints, data sizes, train/test splits, stochastic rollout settings, neural network architectures, optimizer, learning rate, number of epochs, number of heads, QP implementation, and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Each reported metric is aggregated over closed-loop evaluation rollouts with varying initial conditions and stochastic perturbations. The paper specifies the rollout protocol and the sources of variability in the experimental details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix reports the compute setup used for timing experiments and training, including the use of a single NVIDIA GPU, PyTorch implementation, qpth for QP layers, and training/runtime measurements for the evaluated tasks.

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research studies safe robot learning algorithms in simulation and does not involve human subjects, private data, sensitive personal information, or deceptive data collection. The work is intended to improve the reliability and safety of learning-based robotic control systems.

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk pretrained models, generative models, scraped datasets, or assets with substantial misuse risk. The experiments are conducted in simulated robotic-control environments.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the existing simulators, datasets, methods, and software components used in the experiments, including VISTA, OptNet/qpth, BarrierNet, ABNet, DFBNet, and HQP. Publicly available assets are credited through their corresponding papers.

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, or released model asset as a primary contribution. The main contribution is an algorithmic safety-composition framework and its evaluation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology, experiments, and scientific contributions do not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.