LEARNING ADAPTIVE CONTROL FOR SAFE COLLABORATIVE AUTONOMOUS DRIVING

A Thesis

by

FABIAN A. HERNANDEZ

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Computer Science

The University of Texas Rio Grande Valley
May 2026

LEARNING ADAPTIVE CONTROL

FOR SAFE COLLABORATIVE

AUTONOMOUS DRIVING

A Thesis
by
FABIAN A. HERNANDEZ

COMMITTEE MEMBERS

Dr. Qi Lu
Chair of Committee

Dr. Constantine Tarawneh
Committee Member

Dr. Wenjie Dong
Committee Member

Dr. Fatameh Nazari
Committee Member

Dr. Mohamadhossein Noruzoliaee
Committee Member

May 2026

ABSTRACT

Hernandez, Fabian A., Learning Adaptive Control for Safe Collaborative Autonomous Driving. Master of Science (MS), May 2026, 46 pp., 3 tables, 5 figures, 41 references.

Connected autonomous vehicles improve urban driving through collaborative perception via Vehicle-to-Everything (V2X) communication. Frameworks such as V2Xverse leverage this collaboration for planning and perception, yet their controllers rely on fixed parameters that cannot adapt to varying traffic. Control Barrier Functions (CBFs) enforce safety by constraining actions within a safe set, but a fixed barrier gain imposes a single operating point: conservative settings reduce throughput while permissive settings under-react to hazards.

This research proposes an adaptive CBF framework that learns state-dependent, class-specific barrier gains via constrained Reinforcement Learning (RL). A compact policy outputs separate gains for vehicles, pedestrians, and bicycles, parameterizing a CBF quadratic program that modifies only longitudinal control. Training uses constrained PPO with a PID-based Lagrangian multiplier to satisfy a safety cost budget.

We evaluated 315 closed-loop trials across 105 routes in CARLA Town05, comparing the adaptive policy against an unfiltered proportional-integral-derivative (PID) baseline and a fixed-gain CBF filter under identical perception and planning. The adaptive method achieves the highest driving score of 75.81, and reduces collisions per route by 80.3% relative to the unfiltered baseline and by 25.9% relative to the fixed CBF filter, and increases the collision-free route rate to 59.7% while maintaining the best route completion at 90.8%. The greatest improvements appear for vulnerable road users, with bicycle collisions dropping by 87.7% and pedestrian collisions by 54.5% compared to the unfiltered baseline. Because the upstream stack is never modified, all observed improvements are attributable solely to the adaptive safety layer, supporting its portability and ease of integration into other collaborative driving platforms.

DEDICATION

The completion of this thesis would not have been possible without the love and support of family and friends. Thanks to my wonderful mother and sister for always believing in and supporting my goals. Their encouragement, patience, and confidence made each stage of this work possible and meaningful.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Qi Lu, for his guidance and supervision, the support of the MARS (Multiple Autonomous Robot Systems) lab. Additionally, thanks to the thesis committee members Drs. Tarawneh, Dong, Nazari, and Noruzoliaee for their comments, and the financial support provided by the NSF CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (No. 2112650). NSF CISE MSI programs (No. 2318682 & 2431569), and NSF CISE Expand AI Program (No. 2434916).

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Single vehicle autonomy, which relies on camera and LiDAR observations, is fundamentally limited by local sensing, including occlusions, a finite range, and partial observability in dense urban traffic [1, 2, 3]. For example, a pedestrian stepping out from behind a parked truck may be invisible to the ego vehicle but clearly visible to a nearby roadside unit. Connected autonomous vehicles (CAVs) address such limitations by sharing information through Vehicle-to-Everything (V2X) communication [4, 5]. Collaborative perception fuses multi-agent observations to produce richer, more complete representations of the driving environment [6, 7, 8].

V2Xverse [9] advances this line of work by framing collaboration as a full-stack closed-loop driving problem rather than a perception-only benchmark. Its CoDriving pipeline uses collaborative features to improve waypoint prediction and route-level behavior in CARLA [10]. However, stronger perception and planning do not guaranty safe outcomes in themselves. The low-level controller that converts planned waypoints into throttle and brake commands has no explicit model of surrounding actors: it tracks the desired speed faithfully, regardless of whether a vehicle is cutting in or a cyclist is approaching from the side. Safety failures in this stack are therefore a control-level problem, not a perception-level one: collisions occur even when the upstream perception correctly detects all relevant actors.

Control Barrier Functions (CBFs) offer a mechanism to enforce safety at the control level by restricting actions to remain within a forward invariant safe set [11]. A CBF quadratic program (QP) [12, 13] minimally modifies the nominal command to satisfy the barrier constraints. A central parameter is the barrier gain γ, which determines how strongly the filter enforces the safety boundary. Its effect is state-dependent: the same gain can lead to different levels of intervention depending on the current safety margin and relative motion. A fixed γ therefore imposes a single operating point that must simultaneously handle cruise, dense vehicle interac-

tions, pedestrian crossings, and bicycle conflicts. Conservative settings protect safety but reduce throughput; permissive settings preserve progress but under-react to rapidly evolving hazards. This motivates learning a context-dependent gain policy that adapts barrier enforcement online.

We address this by integrating adaptive safety modulation into V2Xverse, employing constrained reinforcement learning to learn state-dependent CBF gains. Building on recent work that learns adaptive barrier gains in simplified multi-agent settings [14], we keep collaborative perception and planning frozen and train only a compact policy ($\sim$10k parameters) that outputs class-specific CBF gains (one each for vehicles, pedestrians, and bicycles) at every decision step. The learned gains parameterize a CBF-QP filter [12] that adjusts longitudinal control while preserving the nominal steering path. Training uses a constrained PPO variant [15] with a PID-based Lagrangian multiplier [16] to satisfy a cumulative safety cost budget. Because the upstream stack is untouched, the training is sample-efficient and any change in safety or efficiency is attributable solely to the adaptive control layer. Unlike end-to-end RL [17] approaches that learn perception, planning, and control jointly, our design is interpretable (the output is a small set of physically meaningful gains), composable (it wraps any nominal controller) and lightweight to train.

The contributions of this thesis are:

- We present an adaptive CBF safety layer integrated into the closed-loop V2Xverse CoDriving stack, adding learned safety modulation without retraining perception or planning.
- We formulate a class-aware adaptive gamma policy trained with constrained PPO and a PID-based Lagrangian that learns state-dependent CBF gains from compact traffic risk features.
- We evaluate on CARLA Town05 under multi-scenario routing, comparing adaptive CBF against fixed CBF and unfiltered baselines under identical perception and planning.
- We show that adaptive CBF modulation reduces collision rates while maintaining route completion, demonstrating that safety can be improved by only learning gain scheduling.

The rest of this thesis is organized as follows. Chapter II reviews related work. Chapter III presents background material. Chapter IV presents the adaptive CBF method. Chapter V describes the experimental setup. Chapter VI reports the experimental results and discusses limitations. Chapter VII concludes.

CHAPTER II

RELATED WORK

2.1 Collaborative Perception and V2X-Aided Driving

Collaborative perception uses vehicle-to-everything (V2X) connectivity to overcome single-agent sensing limits such as occlusions, sparse long-range observations, and limited field of view. Recent surveys categorize information sharing into early, late, and intermediate fusion, highlighting constraints such as communication bandwidth, latency, and synchronization [6, 4, 5]. These works emphasize that message design (what to share) and aggregation (how to fuse) must be co-designed with the downstream task under realistic communication constraints.

A large body of V2V/V2X perception methods focuses on intermediate feature sharing to balance performance and bandwidth. V2VNet [7] employs graph-based aggregation of compressed intermediate representations for joint perception and prediction, targeting occlusion handling through multi-view feature fusion. Communication-aware fusion has also been explored through confidence-guided or spatially selective messaging, where only task-critical spatial regions are transmitted [8]. Complementary to these learning based communication strategies, earlier cooperative 3D perception pipelines explored point cloud feature fusion in edge computing settings, demonstrating the benefits of sharing compact representations over raw sensor streams [18, 19].

Most cooperative perception efforts evaluate open-loop perception metrics, which can understate downstream planning and control consequences. To bridge perception to driving evaluation, Coopernaut [20] studies end-to-end cooperative driving with cross-vehicle message passing, showing that collaboration improves driving outcomes under realistic channel constraints. More recently, V2Xverse [9] advances this direction with a closed-loop, multi-agent simulation platform and an end-to-end collaborative driving system (CoDriving) that integrates communication across the autonomy stack and evaluates system-level driving performance. This distinction is

central to safety research: improving detection does not necessarily translate to fewer collisions without explicit safety mechanisms at the control level.

2.2 Datasets and Benchmarks for Cooperative Perception

Progress in cooperative perception has been accelerated by datasets spanning simulation and real world sensing. OPV2V [21] provides a simulated benchmark with multiple fusion strategies (early/late/intermediate) built on CARLA scenes, enabling controlled study of communication and fusion policies. V2X-Sim [22] similarly offers a large scale collaborative perception dataset and benchmark, supporting standardized evaluation of multi-agent perception pipelines. Real world datasets such as DAIR-V2X [23] expand coverage to vehicle-to-infrastructure cooperative perception with multi-view sensor captures and realistic asynchrony, while V2V4Real [24] targets vehicle-to-vehicle cooperative perception with paired vehicle data and multiple cooperative tasks. In contrast to datasets primarily designed for perception benchmarking, V2Xverse is explicitly designed to support both offline subtask benchmarks and online closed-loop evaluation of collaborative autonomous driving, which is essential for quantifying safety critical outcomes under interaction dynamics [9].

2.3 Control Barrier Functions for Safety Critical Driving

Control Barrier Functions (CBFs) provide a formal mechanism to enforce forward invariance of a safe set by constraining control inputs and have become a standard tool for safety-critical control. The quadratic program (QP) safety filter, widely used, minimally modifies a nominal command to satisfy the barrier constraints [12]. Comprehensive overviews summarize extensions, implementation considerations, and robotics applications, positioning QP-based CBF filtering as a practical interface between learning based planners and safety requirements [11].

Several challenges arise when CBF filters are deployed in autonomous driving. First, the barrier design must account for the limits of actuation and relative degree of the system, motivating variants such as exponential CBF formulations for higher-order dynamics [25, 26]. Second, feasibility can be brittle under modeling error, uncertainty, and multi-agent interactions, often leading to soft constraints (slack variables) or hierarchical formulations that trade off safety and

performance [27]. Third, barrier enforcement depends on coefficients, typically hand-tuned and prone to being overly conservative or insufficiently protective in heterogeneous traffic. Recent work explores adaptive and performance-oriented barrier formulations to improve feasibility and reduce conservatism, including optimal-decay style mechanisms that adjust barrier evolution online [28]. Parallel to these efforts, learning based approaches aim to infer barrier certificates or barrier-like constraints from data, which can reduce manual tuning but often require careful integration to preserve safety guarantees [29, 30, 31].

2.4 Safe Reinforcement Learning for Adaptive Safety Layers

Safe reinforcement learning (RL) is commonly formulated as a constrained Markov decision process (CMDP) [32], where the objective maximizes the expected return while satisfying expected cost constraints. Surveys highlight recurring themes: constraint satisfaction can be enforced through penalty shaping, Lagrangian relaxation, shielding/safety layers, or model-based reachability, each with different sample efficiency and guarantee trade-offs [33]. For on-policy policy-gradient methods, Lagrangian-based updates remain popular due to simplicity and compatibility with scalable deep RL.

Constrained Policy Optimization (CPO) provides trust region style updates that approximately satisfy constraints during learning [34], while PPO offers a clipped surrogate objective that is widely adopted for stable on-policy training [15]. Practical constrained PPO variants often use Lagrangian relaxation, and PID-style updates have been proposed to improve the responsiveness of the Lagrange multiplier under changing constraint violations [16]. Benchmarking environments such as Safety Gym [35] have helped standardize the empirical evaluation of constraint satisfaction in continuous control. In this context, Learning Adaptive Safety for Multi-Agent Systems demonstrates that adaptive CBF parameterization can be learned to balance feasibility, safety, and performance in multi-agent settings [14]. Building on these ideas, the present work focuses on collaborative autonomous driving and adapts CBF enforcement online while keeping perception and planning frozen, targeting system-level safety outcomes under V2X-enabled interaction [9].

CHAPTER III

BACKGROUND

3.1 Kinematic Bicycle Model

The kinematic bicycle model [36] is a standard low-dimensional approximation of vehicle dynamics that captures the essential geometry of front-axle steering while neglecting tire slip and lateral forces. The model collapses the two wheels on each axle into a single contact point and is parameterized by the wheelbase L.

Taking the rear axle as the reference point, the vehicle state is $x = [p_x,\, p_y,\, \theta,\, v]^\top$, where (p_x, p_y) is the position in an inertial frame, θ is the heading angle, and v is the longitudinal speed. The control input is $u = [a,\, \delta]^\top$, where a is the longitudinal acceleration and δ is the front-wheel steering angle.

The continuous-time equations of motion are:

$$\dot{p}_x = v\cos\theta, \qquad \dot{p}_y = v\sin\theta, \qquad \dot{\theta} = \frac{v}{L}\tan\delta, \qquad \dot{v} = a. \tag{3.1}$$

An Euler discretization with time step Δt yields:

$$\begin{aligned} p_{x,k+1} &= p_{x,k} + v_k\cos\theta_k\,\Delta t, \\ p_{y,k+1} &= p_{y,k} + v_k\sin\theta_k\,\Delta t, \\ \theta_{k+1} &= \theta_k + \frac{v_k}{L}\tan\delta_k\,\Delta t, \\ v_{k+1} &= v_k + a_k\,\Delta t, \end{aligned} \tag{3.2}$$

which we write compactly as $x_{k+1} = f(x_k, u_k)$. Because the longitudinal acceleration enters linearly in the speed update, constraints on a can be expressed as linear inequalities, a property that is useful when formulating convex optimization-based controllers.

3.2 Control Barrier Functions

Control barrier functions provide a principled way to enforce safety constraints on dynamical systems [12, 11]. The central idea is to define a scalar function whose sign indicates whether the system is in a safe state, and then to restrict the control input so that this function cannot decrease too quickly.

3.2.1 Extended Class-K Functions

Comparison functions are a standard tool in stability and safety analysis. They formalize the notion of a function that grows with its argument and vanishes at the origin. [37]

Definition 3.1 (Extended class-K function)**.** *A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is an extended class-K function if it is strictly increasing and $\alpha(0) = 0$.*

Unlike standard class-K functions defined only on $[0, \infty)$, extended class-K functions are defined on the entire real line. Common examples include $\alpha(r) = \gamma r$ for $\gamma > 0$ and $\alpha(r) = r^3$. The extended domain is necessary because the barrier value can be negative when the state leaves the safe set.

3.2.2 Safe Sets and Forward Invariance

Consider a discrete-time control system

$$x_{k+1} = f(x_k, u_k), \qquad x_k \in \mathbb{R}^n, \quad u_k \in U \subseteq \mathbb{R}^m, \tag{3.3}$$

where f is the state-transition map and U is the set of admissible inputs.

Given a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, define the safe set C, its boundary ∂C, and its interior as:

$$C = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \qquad \partial C = \{x \in \mathbb{R}^n : h(x) = 0\}, \qquad \mathrm{Int}(C) = \{x \in \mathbb{R}^n : h(x) > 0\}. \tag{3.4}$$

States with $h(x) \geq 0$ are considered safe; states with $h(x) < 0$ are unsafe.

Definition 3.2 (Forward invariance)**.** *A set $C \subseteq \mathbb{R}^n$ is forward invariant for system* (3.3) *under a control law $\mu : \mathbb{R}^n \to U$ if, for every initial condition $x_0 \in C$, the resulting trajectory satisfies $x_k \in C$ for all $k \in \mathbb{N}$.*

Forward invariance captures the safety guarantee: if the system starts in a safe state, it remains safe for all future time. The role of a control barrier function is to certify that such a control law exists.

3.2.3 Discrete-Time Control Barrier Functions

A control barrier function encodes the requirement that, at every state in the safe set, there exists at least one admissible input that prevents the barrier from decreasing faster than a prescribed rate [38, 11].

Definition 3.3 (Discrete-time control barrier function)**.** *A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is a discrete-time control barrier function for system* (3.3) *if there exists a scalar $\gamma \in [0,1]$ such that*

$$\sup_{u_k \in U} \left[h\big(f(x_k,u_k)\big) + (\gamma - 1)h(x_k) \right] \geq 0, \qquad \forall x_k \in C. \tag{3.5}$$

Rearranging, any control satisfying the CBF condition must obey

$$h\big(f(x_k,u_k)\big) - h(x_k) \geq -\gamma h(x_k), \tag{3.6}$$

which bounds how quickly h may decrease in a single step. When $\gamma = 0$ the barrier must be non-decreasing, yielding conservative behavior; when $\gamma = 1$ the condition reduces to $h(x_{k+1}) \geq 0$, permitting faster decay as long as safety is maintained. The linear decay $\gamma h(x_k)$ is a special case of a more general class-$\mathcal{K}$ formulation, but the linear form is most common in practice and is used throughout this work.

Under standard regularity assumptions, selecting controls that satisfy the barrier inequality at each step keeps C forward invariant [38]. In practice, the CBF condition is enforced as a constraint in a quadratic program that minimally modifies a nominal control input.

3.2.4 Adaptive Control Barrier Functions

In the standard CBF formulation, γ is a fixed scalar that governs how quickly the barrier is allowed to decay. The practical effect of γ on constraint tightness depends on the current barrier value. In CBF-QP implementations where the constraint right-hand side contains a term $-\gamma h$, when the barrier is comfortably positive ($h \gg 0$), larger γ makes this term more negative, relaxing the constraint and permitting more aggressive control. However, when operating near the safety boundary where h is small or negative—common in dense traffic—larger γ produces a less negative or even positive contribution, tightening the constraint and enforcing more conservative behavior. A single fixed γ therefore cannot provide the appropriate response across all operating regimes.

An adaptive CBF addresses this by replacing the fixed scalar with a state-dependent function $\gamma(x_k)$:

$$\sup_{u_k \in U} \left[h\big(f(x_k, u_k)\big) + \big(\gamma(x_k) - 1\big) h(x_k) \right] \geq 0, \qquad \forall\, x_k \in C, \tag{3.7}$$

where $\gamma : \mathbb{R}^n \to [0, 1]$ maps the current state to a barrier gain suited to the situation. The function $\gamma(\cdot)$ can be hand-designed or learned from data using reinforcement learning, which is the approach taken in this work [14].

3.3 Safe Reinforcement Learning

Standard reinforcement learning seeks a policy that maximizes the expected cumulative reward. In safety-critical settings, the policy must additionally satisfy constraints that bound undesirable outcomes. This section formalizes the constrained optimization problem that underlies safe RL [32, 33].

3.3.1 Markov Decision Process

A Markov decision process (MDP) is defined by the tuple $(S, A, P, r, \gamma_r, \rho_0)$, where S is the state space, A is the action space, $P(s'|s,a)$ is the transition function, $r(s,a)$ is the reward function, $\gamma_r \in [0,1)$ is the discount factor, and ρ_0 is the initial state distribution. A policy $\pi(a|s)$ maps states to distributions over actions. The RL objective is to find a policy that maximizes the expected discounted return:

$$\max_{\pi} J_R(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma_r^t \, r(s_t, a_t)\right]. \tag{3.8}$$

3.3.2 Constrained Markov Decision Process

A constrained Markov decision process (CMDP) [32] augments the MDP with cost functions $c_i(s,a)$ and corresponding limits d_i. The CMDP seeks a policy that maximizes the return while bounding the expected cumulative cost:

$$\begin{aligned} \max_{\pi} \quad & J_R(\pi), \\ \text{s.t.} \quad & J_{C_i}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma_r^t \, c_i(s_t, a_t)\right] \leq d_i, \quad i = 1, \ldots, m. \end{aligned} \tag{3.9}$$

Separating costs from rewards prevents the policy from trading safety for performance through a single scalar signal. The cost functions encode what should be avoided, while the reward encodes what should be achieved.

3.4 Lagrangian Relaxation and Multiplier Methods

Constrained optimization problems of the form (3.9) are difficult to solve directly with gradient-based methods because the constraints must hold in expectation over trajectories. Lagrangian relaxation recasts the problem as a saddle-point objective over the policy and a nonnegative multiplier, yielding a form amenable to iterative primal-dual methods [34, 16].

3.4.1 Lagrangian Dual

For a single cost constraint $J_C(\pi) \leq d$, the Lagrangian is

$$L(\pi, \lambda) = J_R(\pi) - \lambda \big(J_C(\pi) - d\big), \tag{3.10}$$

where $\lambda \geq 0$ is the Lagrange multiplier. A standard Lagrangian relaxation studies the saddle-point objective

$$\max_{\pi} \min_{\lambda \geq 0} L(\pi, \lambda). \tag{3.11}$$

This objective forms the basis of many constrained policy-gradient methods [34, 16]. When the policy violates the constraint ($J_C > d$), the multiplier grows and penalizes unsafe behavior. When the constraint is satisfied, the multiplier shrinks and the policy is free to focus on reward.

3.4.2 Primal-Dual Updates

The saddle-point objective (3.11) is commonly optimized by alternating between a primal step, which updates the policy parameters to maximize L via policy gradient, and a dual step, which increases the multiplier in proportion to the constraint violation:

$$\lambda \leftarrow \max\big(0,\ \lambda + \eta_\lambda \big(J_C(\pi) - d\big)\big), \tag{3.12}$$

where $\eta_\lambda > 0$ is a step size. This increases λ when the policy is unsafe and decreases it when the policy is overly conservative, providing automatic constraint management.

3.4.3 PID-Based Lagrangian Multiplier

A known limitation of simple dual ascent (3.12) is that the multiplier can oscillate or respond sluggishly, particularly when cost estimates are noisy. The PID Lagrangian [16] replaces the single-step gradient with a proportional-integral-derivative (PID) controller acting on the constraint error $e = \bar{c} - c_{\text{lim}}$, where $\bar{c}$ is the observed mean cost and c_{lim} is the target threshold:

$$\lambda = \text{clip}(k_p\, e + k_i\, I + k_d\, D,\ \ 0,\ \ \lambda_{\max})\,. \tag{3.13}$$

Here k_p, k_i, and k_d are proportional, integral, and derivative gains; I accumulates past errors; and D is a finite-difference derivative. The proportional term provides immediate response, the integral corrects for persistent bias, and the derivative dampens oscillations [39]. Clipping to $[0, \lambda_{\max}]$ ensures the multiplier remains non-negative and bounded.

CHAPTER IV

METHODS

4.1 Problem Formulation and Design Principle

We consider closed-loop collaborative autonomous driving in CARLA, where an ego vehicle must follow a prescribed route while interacting with dynamic traffic participants, including vehicles, pedestrians, and bicycles. The upstream system (collaborative V2X perception and a learned CoDriving planner) provides waypoints and a desired speed at each decision step. Our goal is to improve collision avoidance and maintain route progress without retraining or modifying these upstream modules.

To this end, we introduce a lightweight *safety adapter* that sits between the nominal controller and the actuator. The adapter is a Control Barrier Function (CBF) filter whose barrier gains are dynamically adjusted by a reinforcement learning (RL) policy. Perception, planning, and the nominal steering controller remain unchanged throughout training and evaluation, ensuring that any observed improvements in safety or efficiency can be directly attributed to the adaptive CBF layer.

At each decision step t the deployed stack is:

$$\begin{aligned} &\text{Sensors} \rightarrow \text{Collaborative Perception} \rightarrow \text{Planner} \\ &\rightarrow \text{Nominal Controller} \rightarrow \text{RL-adaptive CBF} \rightarrow \text{Actuation}. \end{aligned} \tag{4.1}$$

CARLA advances in synchronous mode at a fixed physics rate (e.g. 20 Hz), but the agent may skip intermediate frames and reuse the previous control command. RL transitions and CBF optimization are computed only on *decision frames*, so the RL step count corresponds to decision frames rather than raw simulator ticks.

4.2 Nominal Control Interface

Let v_t denote the ego speed and v_t^{des} the desired speed computed by the PID controller from the CoDriving planned waypoints and current target point. We convert this speed target into a nominal longitudinal acceleration:

$$a_t^{\text{nom}} = \frac{v_t^{\text{des}} - v_t}{\Delta t}, \tag{4.2}$$

where Δt is the interval between control updates, not the simulator physics tick. In our experiments, CARLA runs at 20 Hz while control is updated every 4 ticks, so we use $\Delta t = 0.2$ s in the CBF update.

Steering is generated by the nominal PID [40] lateral controller and passed through unchanged; the safety layer modifies only longitudinal behavior. This keeps the optimization lightweight and allows us to isolate the effect of adaptive CBF gains.

4.3 Adaptive CBF Safety Layer

The CBF filter receives the nominal acceleration a_t^{nom}, the current ego state, and the states of nearby traffic participants, and returns a safe acceleration a_t that stays as close to nominal as possible while respecting barrier-based safety constraints. We describe the barrier construction, the quadratic program, and the fallback behavior.

4.3.1 Relative Kinematics and Barrier Definition

For each actor i, we apply a range filter and then a path-aware relevance filter, focusing the CBF on actors that are likely to affect the ego vehicle's near-term trajectory. Let p_{ego} and v_{ego} denote the ego vehicle's position and velocity in the world frame, and p_i and v_i those of actor i. The line-of-sight unit vector from the ego to actor i is:

$$\hat{u}_i = \frac{p_i - p_{\text{ego}}}{\|p_i - p_{\text{ego}}\| + \varepsilon}, \tag{4.3}$$

where $\varepsilon > 0$ is a small constant to prevent division by zero. The Euclidean distance is $d_i = \|p_i - p_{\text{ego}}\|$. The signed relative velocity projected onto this line of sight is:

$$v_i^{\text{los}} = \hat{u}_i^{\top}(v_i - v_{\text{ego}}), \tag{4.4}$$

where a negative value indicates that the actor is approaching the ego. The nonnegative closing speed is:

$$v_i^{\text{close}} = \max\left(0,\ -v_i^{\text{los}}\right). \tag{4.5}$$

Both the ego and each actor are enclosed by a circular footprint whose radius is derived from the CARLA bounding box half-extents as $r = \sqrt{e_x^2 + e_y^2}$, where e_x and e_y are the longitudinal and lateral half-extents of the actor's axis-aligned bounding box. We write r_{ego} for the enclosing radius of the ego vehicle and r_i for that of the actor i. With a geometric margin $m > 0$ and a maximum deceleration parameter of braking a_{brake}, the safety distance and the barrier value are:

$$d_i^{\text{safe}} = r_{\text{ego}} + r_i + m, \tag{4.6}$$

$$h_i = d_i - d_i^{\text{safe}} - \frac{(v_i^{\text{close}})^2}{2\,a_{\text{brake}}}. \tag{4.7}$$

Each term in h_i has a clear physical meaning. The first, d_i, is the current center-to-center distance. The second, d_i^{safe}, covers the physical sizes of both agents plus the clearance margin. The third, $(v_i^{\text{close}})^2/(2\,a_{\text{brake}})$, is the stopping distance that ego would need if it braked at its maximum deceleration a_{brake} from the current closing speed. When $h_i \geq 0$, there is still enough room to stop before reaching the actor i; when $h_i < 0$, the ego is already inside the braking zone and the risk of collision is high. Actors who are far away or moving apart ($v_i^{\text{close}} = 0$) produce large positive h_i and do not constrain the ego.

The path-aware filter is defined in the ego frame. Actors who are sufficiently behind the ego are ignored, and actors far from the ego lane corridor are ignored unless they are likely to cross into the ego path within a short time horizon. This reduces unnecessary braking caused by

traffic in adjacent lanes or in the opposite-direction, while still preserving responsiveness to relevant crossing interactions.

4.3.2 CBF Quadratic Program

At each decision frame, we solve a small Quadratic Program (QP) over the scalar longitudinal acceleration a_t and one slack variable $s_i \geq 0$ per active actor:

$$\min_{a_t, \{s_i\}} \quad (a_t - a_t^{\text{nom}})^2 + w_s \sum_i s_i^2, \tag{4.8}$$

$$\text{s.t.} \quad a_{\min} \leq a_t \leq a_{\max}, \tag{4.9}$$

$$\alpha_i a_t + s_i \geq \beta_i, \quad \forall i. \tag{4.10}$$

The quadratic objective (4.8) keeps the safe acceleration as close to the nominal command as possible, while the penalty weight w_s on slack discourages but does not prohibit soft constraint violations, making the filter robust to infeasible configurations that can arise in dense traffic. Acceleration bounds (4.9) enforce physical actuation limits. Each CBF constraint (4.10) encodes, for actor i, the requirement that the barrier value does not decrease faster than a rate governed by adaptive gain γ_i.

The linearized coefficients are:

$$\alpha_i = -\tfrac{1}{2}\,\eta_i \Delta t^2, \tag{4.11}$$

$$\beta_i = -\gamma_i h_i \; - \; v_i^{\text{los}} \Delta t, \tag{4.12}$$

$$\eta_i = \hat{e}_{\text{ego}}^{\top} \hat{u}_i, \tag{4.13}$$

where $\hat{e}_{\text{ego}}$ is the unit vector along the direction of the ego vehicle and η_i is the direction alignment, defined as the cosine of the angle between the ego's forward direction and the line of sight to actor i. An actor directly ahead has $\eta_i \approx 1$, making the constraint more restrictive; actors behind or to the side have smaller or negative η_i and impose weaker longitudinal restrictions.

An important detail of the implementation is that the barrier h_i uses the nonnegative closing speed v_i^{close} (so receding actors contribute zero braking penalty), while the QP coefficient β_i uses the signed velocity v_i^{los} (so receding actors actively relax the constraint). This asymmetry is deliberate: it prevents the barrier from becoming artificially negative for non-threatening actors while allowing the QP to account for the full relative dynamics correctly.

We solve QP with OSQP[41]. If the solver fails, or if no actor remains after range filtering and path-aware relevance filtering, the nominal command is used directly. After solving, the safe desired speed is obtained from the QP-optimal acceleration a_t^*:

$$v_t^{\text{safe}} = \max(0,\ v_t + a_t^* \Delta t)\,, \tag{4.14}$$

The throttle and brake commands are recomputed from v_t^{safe} via the standard V2Xverse PID speed controller. A detailed derivation of the CBF barrier model and the affine QP form is deferred to the Appendix.

4.4 Adaptive Gamma Policy

A fixed γ imposes a single global safety-performance trade-off in all scenes, which is suboptimal in heterogeneous urban traffic. The core idea of our approach is to have a learned policy choose γ for each actor class at each decision step, conditioned on the current traffic context.

4.4.1 State and Action Spaces

The policy observes an 11-dimensional feature vector summarizing the ego state and the nearest actor of each class:

$$o_t = \big[\, v_t,\ \Delta v_t,\ d_v,\ c_v,\ \eta_v,\ d_p,\ c_p,\ \eta_p,\ d_b,\ c_b,\ \eta_b \,\big], \tag{4.15}$$

where $\Delta v_t = v_t^{\text{des}} - v_t$ is the speed error, and for each actor class $k \in \{v, p, b\}$ (vehicle, pedestrian, bicycle), (d_k, c_k, η_k) are the distance, closing speed and heading alignment of the nearest actor of that class. When no actor of a given class is present, the corresponding entries are set to sentinel values (large distance, zero closing speed, zero alignment). This compact representation captures

the most critical safety information while remaining low-dimensional enough for sample-efficient on-policy learning.

The policy outputs a normalized 3-D action $\tilde{a}_t \in [-1,1]^3$, which is affinely rescaled to class-specific CBF gains:

$$\gamma_t^{(k)} = \frac{\tilde{a}_t^{(k)} + 1}{2} \left(\gamma_{\max}^{(k)} - \gamma_{\min}^{(k)} \right) + \gamma_{\min}^{(k)}. \tag{4.16}$$

Limits $[\gamma_{\min}^{(k)}, \gamma_{\max}^{(k)}]$ are hyperparameters that constrain the policy to a safe operating range.

4.4.2 Network Architecture

Both the actor and critic are implemented as multi-layer perceptrons with two hidden layers of 64 units each and Tanh activations. The actor outputs the mean of a diagonal Gaussian whose log standard deviation is a learnable parameter; the output is passed through a tanh squashing function so that the raw actions lie in $[-1,1]$, and the log-probability is corrected with the standard change-of-variables Jacobian. During training, actions are sampled stochastically for exploration; at evaluation time, the mean is used deterministically. The critic maps the same observation to a scalar state-value estimate. A separate reward critic is used; cost returns are computed as discounted sums directly without a dedicated cost value function.

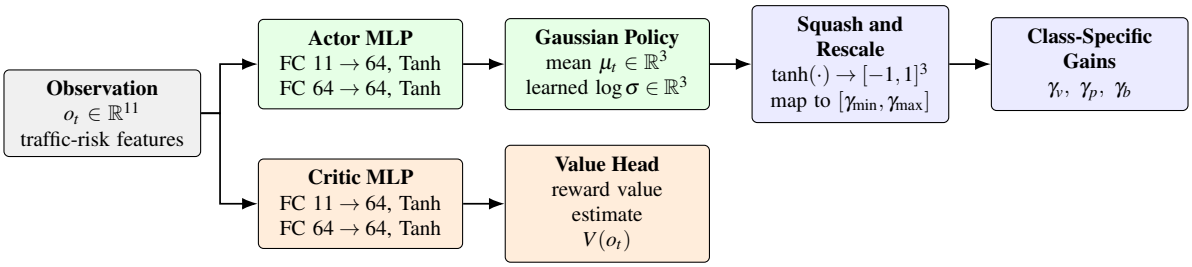


Figure 4.1: Adaptive gamma policy network

4.4.3 Reward Design

The reward encourages the progress of the route. At each decision step, the reward is the nonnegative increase in route completion:

$$r_t^{\text{step}} = \max(0,\ \ell_t - \ell_{t-1})\,, \tag{4.17}$$

where ℓ_t is the cumulative distance traveled along the route by step t. At route termination, an additional completion bonus is added:

$$r_T = r_T^{\text{step}} + \mathbb{I}_{\text{complete}}. \tag{4.18}$$

where $\mathbb{I}_{\text{complete}}$ is 1 if the route was completed and 0 otherwise. The nonnegative clamp prevents the policy from being rewarded for going backward (e.g., during a reversal maneuver). The completion indicator provides a sparse but meaningful signal that the route was finished successfully, encouraging the policy not to be so conservative that it stalls.

4.4.4 Safety Cost

Rather than folding safety into the reward (which conflates two competing objectives), we define a separate per-step cost that feeds into the constrained optimization. For each actor i in the safety range, two risk measures are calculated.

Proximity cost. A piecewise-linear function of distance that ramps from 0 at an outer warning threshold to 1 at the collision envelope:

$$c_i^{\text{prox}} = \begin{cases} 0, & d_i \geq d_{\text{warn}}, \\ \dfrac{d_{\text{warn}} - d_i}{d_{\text{warn}} - d_{\text{col}}}, & d_{\text{col}} < d_i < d_{\text{warn}}, \\ 1, & d_i \leq d_{\text{col}}, \end{cases} \tag{4.19}$$

where d_{warn} is a proximity warning distance beyond which no cost accrues (distinct from the CBF safe distance d_i^{safe}), and d_{col} is the approximate collision-contact distance.

Time-to-collision cost. A linear ramp based on how soon contact would occur at the current closing speed:

$$\mathrm{TTC}_i = \frac{d_i}{v_i^{\mathrm{close}} + \varepsilon}, \tag{4.20}$$

$$c_i^{\mathrm{ttc}} = \max\left(0,\, 1 - \frac{\mathrm{TTC}_i}{\tau_{\mathrm{ttc}}}\right). \tag{4.21}$$

Here τ_{ttc} is a time-to-collision horizon; the cost is zero when $\mathrm{TTC}_i \geq \tau_{\mathrm{ttc}}$ (actor receding or far away) and increases toward 1 as imminent contact approaches.

The cost of decision-step aggregates across all actors by taking the worst case:

$$c_t^{\mathrm{step}} = \max_i \left\{ \max\left(c_i^{\mathrm{prox}},\, c_i^{\mathrm{ttc}}\right) \right\}. \tag{4.22}$$

At route termination, a discrete collision penalty is added:

$$c_T = c_T^{\mathrm{step}} + c_{\mathrm{col}}\, \mathbb{I}_{\mathrm{collision}}, \tag{4.23}$$

where c_{col} is a configurable terminal penalty and $\mathbb{I}_{\mathrm{collision}}$ is equal to 1 if a collision occurred during the route and 0 otherwise. The dense per-step cost provides a continuous learning signal for the policy to preemptively avoid risky states, while the terminal penalty ensures that actual collisions are heavily penalized even if they occur from states that appeared safe one step earlier.

4.5 Constrained Policy Optimization

We train the adaptive gamma policy using a PPO-style algorithm extended with a Lagrangian constraint on the cumulative safety cost. The total loss is:

$$\mathscr{L} = \mathscr{L}_{\text{reward}} + \lambda\, \mathscr{L}_{\text{cost}} - \alpha_H\, \mathscr{H}[\pi] + \alpha_V\, \mathscr{L}_{\text{value}}, \tag{4.24}$$

where α_H is the entropy bonus coefficient, α_V is the value-function loss coefficient, $\mathscr{H}[\pi]$ is the policy entropy (encouraging exploration), $\mathscr{L}_{\text{value}}$ is the loss of the reward-critic regression, and λ is a dynamically adjusted Lagrangian multiplier that controls the weight of the safety cost relative to reward.

4.5.1 Clipped Surrogates

Let $\rho_t = \pi_\theta(a_t|o_t) \,/\, \pi_{\theta_{\text{old}}}(a_t|o_t)$ be the importance ratio and $\varepsilon \in (0,1)$ the PPO clipping coefficient. The reward and cost policy-gradient surrogates are as follows:

$$\mathscr{L}_{\text{reward}} = -\mathbb{E}[\min(\rho_t A_t,\ \text{clip}(\rho_t,\ 1-\varepsilon,\ 1+\varepsilon) A_t)], \tag{4.25}$$

$$\mathscr{L}_{\text{cost}} = \quad \mathbb{E}[\max(\rho_t C_t,\ \text{clip}(\rho_t,\ 1-\varepsilon,\ 1+\varepsilon) C_t)], \tag{4.26}$$

where A_t are GAE-based reward advantages (normalized to zero mean and unit variance) and C_t are discounted cumulative cost returns computed directly without a cost baseline and left unnormalized. The asymmetry in clipping direction is deliberate: the min in $\mathscr{L}_{\text{reward}}$ is standard PPO (preventing large reward-increasing updates), while the max in $\mathscr{L}_{\text{cost}}$ is conservative and takes the larger of the clipped and unclipped cost surrogate, preventing the optimizer from under-counting safety violations through ratio clipping.

4.5.2 PID Lagrangian Update

The multiplier λ is updated after each rollout batch using a PID controller. Let j denote the rollout-update index, let $\bar{c}_j$ denote the mean per-step safety cost in update j, and let c_{limit} be the target cost threshold. The constraint error is:

$$e_j = \bar{c}_j - c_{\text{limit}}. \tag{4.27}$$

The multiplier is:

$$\lambda_j = \text{clip}\big(k_p e_j + k_i I_j + k_d D_j,\ 0,\ \lambda_{\max}\big), \tag{4.28}$$

where k_p, k_i, and k_d are proportional, integral, and derivative gains, I_j is the running integral of past constraint errors, and D_j is a finite-difference derivative term computed from recent batch-cost statistics. The PID formulation provides smoother multiplier dynamics than a simple dual gradient ascent, which is particularly important in our setting where rollout cost estimates are noisy due to the stochasticity of CARLA traffic scenarios.

4.6 Training and Evaluation Protocol

Training is fully on-policy: the adaptive-gamma policy collects rollouts in CARLA, updates via the constrained PPO loss, and repeats. Collaborative perception, the CoDriving planner, and the nominal steering controller remain unchanged throughout; only the small actor-critic network is optimized. This makes training lightweight and ensures that any change in driving behavior is attributable solely to adaptive safety modulation. For evaluation, we load the trained policy checkpoint and select actions deterministically (using the actor mean without sampling noise). All other components are identical to training, which allows for fair comparison against fixed-CBF and unfiltered baselines.

CHAPTER V

EXPERIMENTAL SETUP AND DESIGN

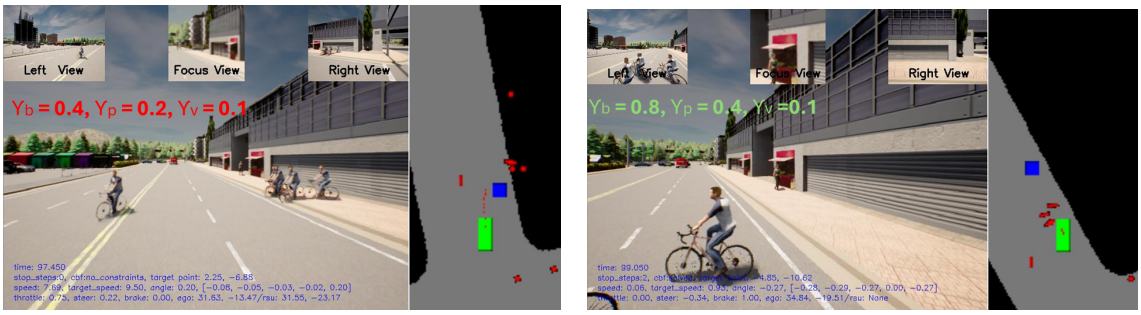
5.1 Simulation Setup

All experiments run in CARLA 0.9.10.1 on Town05, an urban map with multi-lane roads, signalized and unsignalized intersections, and mixed traffic. Evaluation uses a fixed set of predefined CARLA waypoint routes covering straights, turns, lane merges, and pedestrian/cyclist interactions; the same route set is used for every method. We use the V2Xverse CoDriving pipeline with the collaborative perception and planning checkpoints frozen. The ego agent decides every 4 physics ticks (5 Hz, Δt=0.2 s). The planner outputs waypoints and a desired speed capped at 10 m/s. A PID lateral controller tracks the waypoints unchanged; the safety layer only modifies the desired longitudinal speed.

Fig. 5.1 illustrates the adaptive CBF in action during a typical Town05 encounter with cyclists. As the ego vehicle detects bicycles ahead near the sidewalk, the policy assigns γ_b=0.4, γ_p=0.2, and γ_v=0.1, keeping a moderate bicycle gain while leaving the vehicle and pedestrian gains low, allowing the ego vehicle to continue at reduced speed. As the cyclist approaches the ego's path, the policy changes the bicycle gain to γ_b=0.8 and the pedestrian gain to γ_p=0.4 while the vehicle gain stays at γ_v=0.1, and the CBF filter brings the vehicle to a full stop until the actor clears the path.

5.2 CBF Filter Parameters

Table 5.1 lists the CBF filter parameters. Path-aware gating discards actors behind the rear threshold or outside the lateral corridor, unless their estimated crossing TTC is below the threshold. The speed-dependent corridor term prevents false braking for adjacent-lane traffic.



(a) (b)

Figure 5.1: Adaptive barrier gains during a cyclist encounter. (a) Bicycles detected ahead, speed 7.69 m/s. (b) Cyclist in the path, so the vehicle stopped.

Table 5.1: CBF filter hyperparameters.

Parameter	**Symbol**	**Value**
Max actor range	$d_{\max}$	30.0 m
Safety margin	m	2.5 m
Braking deceleration	a_{brake}	6.0 m/s^2
Min / max acceleration	$a_{\min}, a_{\max}$	-8.0 / 4.0 m/s^2
Slack penalty	w_s	1000
Control interval	Δt	0.2 s
Path-aware actor gating		
Rear ignore distance		2.0 m
Lateral corridor half width		2.5 m + 0.04 v_{ego}
Crossing TTC threshold		3.2 s

5.3 RL Training Configuration

The adaptive gamma policy is trained with constrained PPO (Section 4). Training is on-policy: each iteration collects 2000 decision steps, then performs 10 epochs of minibatch PPO updates with PID-based Lagrangian adjustment. A small entropy bonus (α_H=0.005) aids early exploration. The actor-critic has $\sim$10k parameters; a full training run of 800k–1M environment steps takes 3–5 days on a single NVIDIA RTX 6000 Ada GPU. Table 5.2 lists all hyperparameters.

Table 5.2: RL training hyperparameters.

Parameter	**Value**
Obs / action dim	11 / 3 (one γ per class)
Hidden layers	2×64, Tanh
Learning rate	3×10^{-4}
γ_{RL} / GAE λ	0.99 / 0.95
PPO clip ε	0.2
Epochs / minibatches	10 / 8
Rollout size	2000 steps
α_H / α_V / grad clip	0.005 / 0.5 / 0.5
Constrained optimization	
Cost limit c_{limit}	0.1
PID (k_p, k_i, k_d) / EMA α	(1.0, 0.1, 0.1) / 0.9
$\lambda_{\max}$	100.0
c_{col} (terminal)	3.0
d_{warn} / d_{col} / τ_{ttc}	10.0 m / 2.5 m / 3.0 s
$[\gamma_{\min}, \gamma_{\max}]$ all classes	[0.05, 0.8]

5.4 Baselines

We compare three longitudinal control configurations that share frozen perception, planning, and lateral control:

No CBF: Raw planner output passes to the unfiltered PID-only actuator (unmodified V2Xverse stack).

Fixed CBF: CBF-QP with constant $\gamma_v{=}\gamma_p{=}\gamma_b{=}0.5$, selected based on preliminary experiments that indicated that this value balanced safety and route completion.

Adaptive CBF: CBF-QP with class-specific γ values in $[0.05, 0.8]$ selected per step by the trained RL policy. The evaluation checkpoint is the best saved checkpoint selected by the highest driving score, with ties broken by fewer collisions per route and then higher route completion.

5.5 Evaluation Protocol and Metrics

All methods are evaluated in deterministic mode (mean policy action) on the same Town05 routes with fixed random seeds. Multiple passes per method account for the residual nondeterminism of CARLA. We report the CARLA driving score (route completion penalized by infractions), the route score (mean percentage of route completed), the route completion rate (fraction of trials where the full route was finished), collisions per route and per kilometer (split by actor class: vehicle, bicycle, pedestrian, static) and the collision-free route fraction. A method is considered an improvement if it reduces collision rates without a proportional loss in route completion.

CHAPTER VI

EXPERIMENTAL RESULTS

All three methods are evaluated over 315 route trials on Town05. Table 6.1 summarizes the aggregate metrics, while Figs. 6.1–6.3 visualize the score distributions and per-class collision breakdown.

Table 6.1: Aggregate evaluation results on Town05 (315 trials each; driving score shown with 95% CI). Collision columns are split by actor class: vehicles, bicycles, and pedestrians.

Method	**Driving Score** ↑	**Route Score** ↑	**Route Compl.** ↑	**Coll. /Route** ↓	**Coll. /km** ↓	**Coll. Free** ↑	**Veh. /Rt** ↓	**Bike /Rt** ↓	**Ped. /Rt** ↓	**Static /Rt** ↓
PID-only	45.47 (±4.41)	96.91	90.2%	2.952	35.58	31.7%	0.968	1.527	0.390	0.067
Fixed CBF	67.39 (±3.58)	95.36	88.3%	0.784	9.56	48.6%	0.168	0.286	0.302	0.029
Adaptive CBF	**75.81** (±3.35)	96.71	**90.8%**	**0.581**	**6.75**	**59.7%**	0.187	**0.187**	**0.178**	0.029

6.1 Safety Analysis

As shown in Table 6.1, the adaptive CBF policy scores 75.81 ($\pm$ 3.35) on the driving metric and produces the fewest collisions across bicycles and pedestrians. Relative to PID-only, total collisions per route drop by 80.3% (2.952 to 0.581), and collisions per kilometer by 81.0% (35.58 to 6.75), with bicycles seeing the steepest reduction at 87.7% (1.527 to 0.187). Vehicle and pedestrian collisions fall by 80.7% (0.968 to 0.187) and 54.5% (0.390 to 0.178), respectively.

The adaptive policy further improves on the fixed CBF baseline, cutting collisions per route by 25.9% (0.784 to 0.581) and collisions per kilometer by 29.3% (9.56 to 6.75). Most of this improvement comes from pedestrians (41.1% lower) and bicycles (34.4% lower), which aligns with the design intent: the RL policy changes γ_p and γ_b when vulnerable road users approach the ego trajectory, braking more aggressively only when it matters. The collision-free route rate rises from 48.6% under fixed CBF to 59.7% under the adaptive CBF. Fig. 6.1 shows the per-class breakdown. PID-only's total is dominated by bicycle collisions, which account for over half of its bar. Both CBF methods reduce bicycle and vehicle collisions substantially, bring-

ing those categories to comparable levels. The adaptive policy then separates itself from the fixed variant primarily through lower pedestrian collisions, consistent with its ability to adapt γ when vulnerable road users are nearby.

6.2 Efficiency Analysis

Despite being more conservative on safety, the adaptive policy does not sacrifice route progress. It slightly edges out PID-only route completion (90.8% vs. 90.2%), while fixed CBF completes only 88.3%. The fixed gain tends to over-brake in tight gaps, stalling the vehicle; the adaptive policy avoids this by selecting context-dependent gains when no hazard is nearby, so it records only 24 blocked-agent failures versus 35 for fixed CBF and 31 for PID-only. The penalty factor (driving score divided by route score) captures how cleanly routes are completed: 0.784 for adaptive policy, 0.707 for fixed CBF, and 0.469 for PID-only. In other words, PID-only loses more than half its route score to infraction penalties on an average run.

6.3 Score Distribution

The violin plots in Fig. 6.2 give a fuller picture than the averages alone. The black line represents the mean, and the red line represents the median. More than half of the adaptive CBF routes (56.2%) score a perfect 100, pulling the median to 100.0. Fixed CBF sits at a median of 60.0 and PID-only at 30.0; the latter shows a heavy lower tail with 85 routes below a score of 10, each flagging an early collision that dominates the final penalty. The tighter interquartile range for adaptive policy ($[50, 100]$ versus $[8, 100]$ for PID-only) suggests that improvement holds in a wide variety of route layouts rather than being concentrated on a few easy ones. Route-level paired comparisons support this: the adaptive policy outscores PID-only on 64.4% of routes (worse on only 7.9%) and beats fixed CBF on 36.2% (worse on 18.7%).

6.4 Infraction Breakdown

Fig. 6.3 breaks down the mean infractions per route in the five categories shown. Bicycle infractions are the largest for PID-only at 1.527 per route; fixed CBF reduces this to 0.286, and the adaptive policy to 0.187. Vehicle infractions follow a similar trend: 0.968 for PID-only, 0.168 for fixed CBF, and 0.187 for adaptive. Pedestrian infractions drop from 0.390 (PID-only) to 0.302 (fixed CBF) and 0.178 (adaptive). Static object infractions are low across the board – 0.067, 0.029, and 0.029 for PID-only, fixed CBF, and adaptive, respectively. Stop-sign infractions round out the picture at 0.248 for PID-only, 0.235 for fixed CBF, and 0.168 for the adaptive policy.

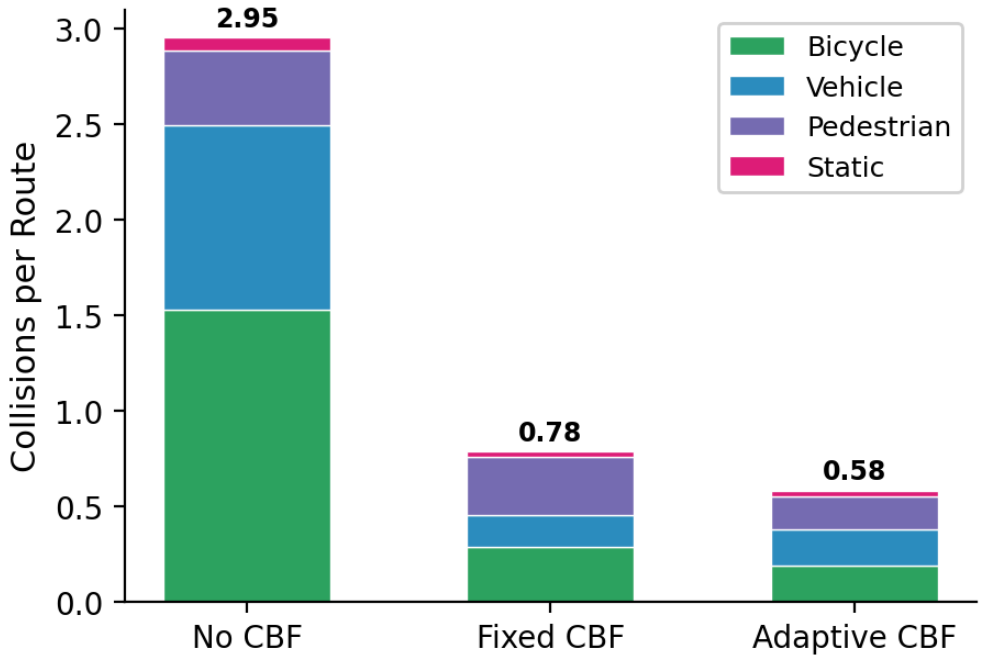


Figure 6.1: Mean collisions per route, stacked by actor class.

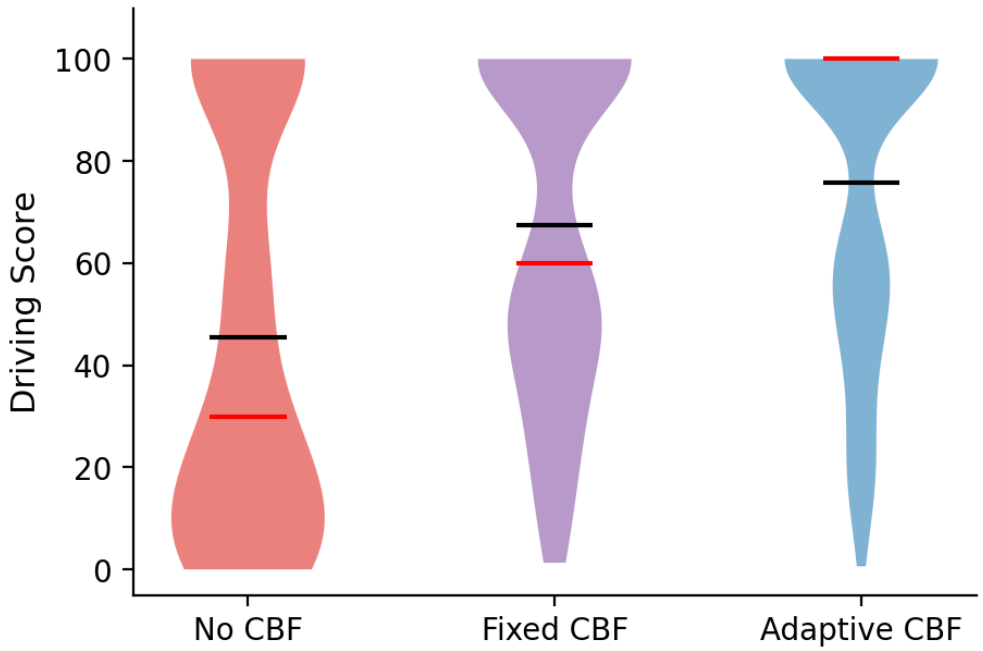


Figure 6.2: Per-route driving score distributions

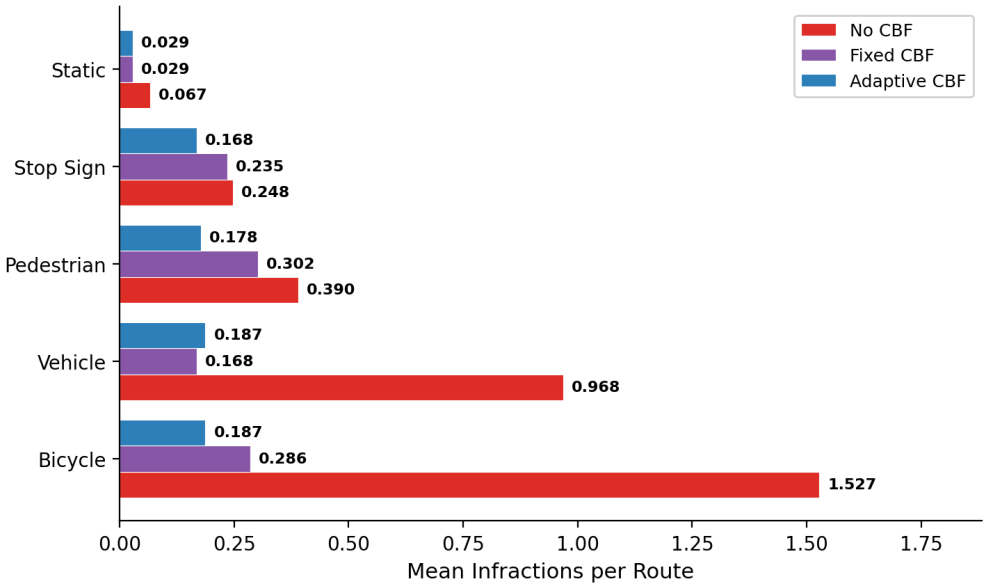


Figure 6.3: Mean infractions per route by category.

6.5 Discussion and Limitations

The evaluation is conducted entirely in CARLA using the V2Xverse closed-loop stack. The implementation used in this thesis is available at `https://github.com/MARSLab-UTRGV/V2Xverse-ASRL`. V2Xverse itself supports robustness studies for communication bandwidth, pose error, and latency, but those settings are not varied in the experiments reported here. Instead, this thesis uses the nominal closed-loop configuration with frozen perception and planning, so the collaborative inputs are treated as available at each decision frame and the comparison isolates the control-layer question: whether adaptive CBF gain scheduling can improve safety under the same upstream stack. Future work should evaluate the same safety layer under delayed, dropped, bandwidth-limited, or noisy cooperative messages to measure robustness to realistic communication failures.

The method also combines two different notions of constraint handling. The reinforcement learning problem uses a Lagrangian relaxation, which treats safety cost as an expected constraint during training rather than as a hard per-step guarantee. In contrast, the deployed CBF-QP acts online at every decision frame and directly constrains the longitudinal acceleration. Thus, the RL policy does not replace the CBF safety filter; it only selects the class-specific barrier gains used by the filter. At the same time, the implemented QP includes slack variables to maintain feasibility in dense traffic, so the practical system should be interpreted as a soft-constrained safety filter rather than a formal hard-invariance guarantee.

Future work will investigate validation on a small-scale connected autonomous vehicle testbed. Such a system would require onboard localization, low-latency communication between vehicles or infrastructure nodes, synchronized state estimation, and a real-time implementation of the CBF-QP safety layer. A scaled platform would make it possible to study effects that are abstracted away in simulation, including sensing noise, actuation delay, communication dropouts, and calibration error. The modular design of this work is well suited to that transition because the adaptive policy and CBF filter can be placed between an existing planner and low-level vehicle controller.

CHAPTER VII

CONCLUSION

This thesis introduces an adaptive CBF safety layer for collaborative autonomous driving that works with the V2Xverse CoDriving stack without retraining perception or planning. By adapting only the safety enforcement while keeping upstream modules fixed, our approach ensures that observed improvements are attributable to the safety layer, suggesting its portability and ease of integration into other collaborative driving platforms.

The approach combines a modular CBF-QP safety filter with a class-aware adaptive γ policy trained using constrained PPO and a PID-based Lagrangian. A compact network outputs gains per-step for vehicles, pedestrians, and bicycles from traffic-risk features, while the CBF remains the final mechanism that enforces bounded longitudinal actions. This keeps the safety layer interpretable, lightweight, and directly compatible with an existing collaborative autonomy stack.

When evaluated in Town05 against PID-only and fixed CBF under the same perception, planning, and steering setup, the adaptive method delivered the strongest overall performance: driving score 75.81 ± 3.35, collisions per route 0.581, and collisions per kilometer 6.75. Relative to PID-only, collisions per route dropped by 80.3%, and relative to fixed CBF, they dropped by 25.9%. The completion of the route remained high at 90.8%, while the collision-free route rate increased to 59.7%. The greatest gains on bicycles and pedestrians show that adaptive gain scheduling is especially useful for vulnerable road users in heterogeneous traffic.

Overall, these results support the central claim of this work: learning to schedule barrier gains online can improve safety without sacrificing route progress. More broadly, they show that meaningful safety improvements can come from a targeted control-layer intervention rather than a full redesign of the autonomy stack.

The current safety layer modifies only longitudinal control. Extending the framework to jointly regulate longitudinal and lateral actions is a natural next step, though it requires a coupled vehicle model and richer barrier construction. Even with this limitation, the approach is practical as a drop-in safety adapter for existing autonomy stacks and complements continued advances in collaborative perception and planning.

REFERENCES

[1] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. M. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451, 2019.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[3] E. Marti, M. A. de Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[4] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, "Collaborative perception in autonomous driving: Methods, datasets, and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 6, pp. 131–151, 2023.

[5] T. Huang, J. Liu, X. Zhou, D. C. Nguyen, M. Rahimi Azghadi, Y. Xia, Q.-L. Han, and S. Sun, "Vehicle-to-everything cooperative perception for autonomous driving," *Proceedings of the IEEE*, vol. 113, no. 5, pp. 443–477, 2025.

[6] S. Teufel, J. Gamerdinger, J.-P. Kirchner, G. Volk, and O. Bringmann, "Collective perception datasets for autonomous driving: A comprehensive review," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 1548–1555.

[7] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 605–621.

[8] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: communication-efficient collaborative perception via spatial confidence maps," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[9] G. Liu, Y. Hu, C. Xu, W. Mao, J. Ge, Z. Huang, Y. Lu, Y. Xu, J. Xia, Y. Wang, and S. Chen, "Toward collaborative autonomous driving: Simulation platform and end-to-end system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 8, pp. 6566–6584, 2025.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[11] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[13] A. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," *53rd IEEE Conference on Decision and Control*, pp. 6271–6278, 2014.

[14] L. Berducci, S. Yang, R. Mangharam, and R. Grosu, "Learning adaptive safety for multi-agent systems," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 2859–2865.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *openai*, 2017.

[16] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by pid lagrangian methods," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[17] W. Jun, Y.-J. Won, P. Park, K. Chun, and S. Lee, "Performance analysis of rl based end-to-end learning technology for autonomous driving," in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, 2025, pp. 1–5.

[18] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, ser. SEC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 88–100.

[19] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 514–524.

[20] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[21] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2583–2589.

[22] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 914–10 921, 2022.

[23] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 21 329–21 338.

[24] R. Xu, X. Xia, J. Li, H. Li, S. Zhang, Z. Tu, Z. Meng, H. Xiang, X. Dong, R. Song, H. Yu, B. Zhou, and J. Ma, "V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 712–13 722.

[25] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328.

[26] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[27] J. J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," in *Proceedings of Robotics: Science and Systems (RSS)*, vol. abs/2004.07584, 2020.

[28] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier functions with guaranteed point-wise feasibility," 2021.

[29] S. Prajna and A. Rantzer, "On the necessity of barrier certificates," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 526–531, 2005, 16th IFAC World Congress.

[30] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 120. PMLR, 10–11 Jun 2020, pp. 708–717.

[31] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," in *International Conference on Learning Representations (ICLR)*, 2021.

[32] E. Altman, *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

[33] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, p. 1437–1480, Jan. 2015.

[34] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 22–31.

[35] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," 2019.

[36] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 812–818.

[37] G. Bitsoris, *Comparison and Stability*. Cham: Springer Nature Switzerland, 2025, pp. 203–279. [Online]. Available: https://doi.org/10.1007/978-3-031-76220-8_6

[38] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[39] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, and L. Zhang, "A pid controller approach for stochastic optimization of deep networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8522–8531.

[40] K. Bensassi, T. Jarou, J. Abdouni, M. Amkrane, O. Id-Ouakrim, and M. Abarkan, "Control strategies for autonomous vehicles: A comparative review of classical and intelligent methods," *2025 International Conference on Electrical Systems & Automation (ICESA)*, pp. 1–6, 2025.

[41] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

APPENDIX

APPENDIX

DERIVATION OF THE BARRIER FUNCTION AND LINEARIZED QP COEFFICIENT

This appendix derives the barrier function in Eq. (4.7) and the linearized coefficients used in the quadratic program.

A.1 Relative Kinematics

For actor i, define the relative position vector

$$\rho_i = p_i - p_{\text{ego}} \in \mathbb{R}^2, \tag{A.1.1}$$

with Euclidean distance

$$d_i = \|\rho_i\|. \tag{A.1.2}$$

For the derivation below, assume $d_i > 0$ and define the line-of-sight (LOS) unit vector as

$$\hat{u}_i = \frac{\rho_i}{\|\rho_i\|}. \tag{A.1.3}$$

In the implementation, this direction is regularized numerically as

$$\hat{u}_i \approx \frac{\rho_i}{\|\rho_i\| + \varepsilon}, \tag{A.1.4}$$

with $\varepsilon > 0$ a small constant to avoid division by zero.

The relative velocity is

$$v_i^{\text{rel}} = v_i - v_{\text{ego}}. \tag{A.1.5}$$

The LOS-projected relative velocity is then

$$v_i^{\text{los}} = \hat{u}_i^\top v_i^{\text{rel}}. \tag{A.1.6}$$

This quantity is the rate of change of the distance d_i. To show this, note that

$$d_i = \|\rho_i\| = \sqrt{\rho_i^\top \rho_i}. \tag{A.1.7}$$

Differentiating with respect to time gives

$$\begin{aligned}
\dot{d_i} &= \frac{d}{dt}\sqrt{\rho_i^\top \rho_i} && \text{(A.1.8)}\\
&= \frac{1}{2}(\rho_i^\top \rho_i)^{-1/2}\frac{d}{dt}(\rho_i^\top \rho_i) && \text{(A.1.9)}\\
&= \frac{1}{2\|\rho_i\|}\cdot 2\rho_i^\top \dot{\rho}_i && \text{(A.1.10)}\\
&= \frac{\rho_i^\top \dot{\rho}_i}{\|\rho_i\|}. && \text{(A.1.11)}
\end{aligned}$$

Since $\dot{\rho}_i = v_i - v_{\text{ego}} = v_i^{\text{rel}}$, we obtain

$$\dot{d_i} = \frac{\rho_i^\top}{\|\rho_i\|}(v_i - v_{\text{ego}}) = \hat{u}_i^\top (v_i - v_{\text{ego}}) = v_i^{\text{los}}. \tag{A.1.12}$$

Hence, $v_i^{\text{los}} < 0$ means the actor is approaching the ego, while $v_i^{\text{los}} > 0$ means the separation is increasing.

We define the nonnegative closing speed as

$$v_i^{\text{close}} = \max(0, -v_i^{\text{los}}). \tag{A.1.13}$$

A.2 Derivation of the Barrier Function

The safety distance is chosen as

$$d_i^{\text{safe}} = r_{\text{ego}} + r_i + m, \tag{A.2.1}$$

where r_{ego} and r_i are the enclosing radii of the ego and actor footprints, and $m > 0$ is a geometric margin.

The quantity

$$\frac{(v_i^{\text{close}})^2}{2a_{\text{brake}}} \tag{A.2.2}$$

is the classical one-dimensional constant-deceleration stopping distance. Indeed, for a particle with initial speed v_0 and constant braking magnitude $a_{\text{brake}} > 0$, the kinematic relation

$$v^2 = v_0^2 + 2a\Delta x \tag{A.2.3}$$

with final speed $v = 0$ and acceleration $a = -a_{\text{brake}}$ yields

$$0 = v_0^2 - 2a_{\text{brake}}\Delta x \quad \Longrightarrow \quad \Delta x = \frac{v_0^2}{2a_{\text{brake}}}. \tag{A.2.4}$$

Substituting $v_0 = v_i^{\text{close}}$ gives the stopping-distance term used by the filter.

Therefore, the barrier

$$h_i = d_i - d_i^{\text{safe}} - \frac{(v_i^{\text{close}})^2}{2a_{\text{brake}}} \tag{A.2.5}$$

has the following interpretation:

- d_i is the current center-to-center separation,
- d_i^{safe} is the geometric collision buffer,
- $(v_i^{\text{close}})^2/(2a_{\text{brake}})$ is the additional distance suggested by a one-dimensional constant-braking approximation.

A.3 Discrete-Time CBF Condition

Let x_t denote the current joint state of the ego and actor, and let a_t denote the longitudinal acceleration selected by the QP. We impose the discrete-time CBF condition

$$h_i(x_{t+1}) - h_i(x_t) \geq -\gamma_i h_i(x_t), \tag{A.3.1}$$

which is equivalent to

$$h_i(x_{t+1}) \geq (1-\gamma_i) h_i(x_t). \tag{A.3.2}$$

When $0 < \gamma_i \leq 1$, this condition limits how quickly the barrier is allowed to decrease from one decision step to the next.

A.4 One-Step Linearization

To obtain an affine constraint in the control input, we linearize the one-step barrier update. Let Δt denote the decision-frame interval and let $\hat{e}_{\text{ego}}$ denote the ego heading unit vector,

$$\hat{e}_{\text{ego}} = \begin{bmatrix} \cos\psi_{\text{ego}} \\ \sin\psi_{\text{ego}} \end{bmatrix}. \tag{A.4.1}$$

Define the heading alignment as the cosine of the angle between the ego's forward direction and the line of sight to actor i:

$$\eta_i = \hat{e}_{\text{ego}}^{\top} \hat{u}_i. \tag{A.4.2}$$

Over one decision step, the distance update can be decomposed into a velocity-driven term and an acceleration-driven term:

$$d_i(t+\Delta t) \approx d_i(t) + \dot{d}_i(t)\Delta t + \Delta d_i^{(a)}, \tag{A.4.3}$$

where $\dot{d}_i(t) = v_i^{\text{los}}$ is the rate of change of distance derived earlier. The term $\Delta d_i^{(a)}$ captures the contribution from ego acceleration and arises from the standard kinematic relation for displacement under constant acceleration, $\frac{1}{2}a\Delta t^2$, projected onto the line of sight.

Under a constant longitudinal acceleration a_t over one control interval, the ego displacement induced by acceleration is

$$\Delta p_{\text{ego}}^{(a)} \approx \frac{1}{2} a_t \Delta t^2 \hat{e}_{\text{ego}}. \tag{A.4.4}$$

Since the relative position is $\rho_i = p_i - p_{\text{ego}}$, this contributes

$$\Delta \rho_i^{(a)} \approx -\frac{1}{2} a_t \Delta t^2 \hat{e}_{\text{ego}}. \tag{A.4.5}$$

Projecting this change onto the LOS direction yields the corresponding change in distance:

$$\Delta d_i^{(a)} \approx \hat{u}_i^\top \Delta \rho_i^{(a)} \tag{A.4.6}$$

$$= -\frac{1}{2} a_t \Delta t^2 \hat{u}_i^\top \hat{e}_{\text{ego}} \tag{A.4.7}$$

$$= -\frac{1}{2} \eta_i \Delta t^2 a_t. \tag{A.4.8}$$

Therefore,

$$d_i(t + \Delta t) \approx d_i(t) + v_i^{\text{los}} \Delta t - \frac{1}{2} \eta_i \Delta t^2 a_t. \tag{A.4.9}$$

Under this local approximation, the barrier update is dominated by the distance update in (A.4.9), so

$$h_i(x_{t+1}) \approx h_i(x_t) + v_i^{\text{los}} \Delta t - \frac{1}{2} \eta_i \Delta t^2 a_t. \tag{A.4.10}$$

A.5 Derivation of the QP Coefficients

Substituting (A.4.10) into the discrete-time CBF condition (A.3.1) gives

$$v_i^{\text{los}}\Delta t - \frac{1}{2}\eta_i \Delta t^2 a_t \geq -\gamma_i h_i(x_t). \tag{A.5.1}$$

Rearranging terms yields

$$-\frac{1}{2}\eta_i \Delta t^2 a_t \geq -\gamma_i h_i(x_t) - v_i^{\text{los}}\Delta t. \tag{A.5.2}$$

Hence the affine constraint can be written as

$$\alpha_i a_t \geq \beta_i, \tag{A.5.3}$$

with

$$\alpha_i = -\frac{1}{2}\eta_i \Delta t^2, \tag{A.5.4}$$

$$\beta_i = -\gamma_i h_i(x_t) - v_i^{\text{los}}\Delta t. \tag{A.5.5}$$

Including a nonnegative slack variable $s_i \geq 0$ to soften infeasible constraints gives the implemented form

$$\alpha_i a_t + s_i \geq \beta_i. \tag{A.5.6}$$

This completes the derivation of the linearized coefficients used in the CBF-QP.

VITA

Fabian A. Hernandez earned a Bachelor of Science degree in Computer Engineering, with a minor in Applied Mathematics, in December 2023. He further pursued his academic interests by earning a Master of Science in Computer Science from The University of Texas Rio Grande Valley in May 2026. Throughout his academic career, he maintained a 4.0 GPA and graduated Summa Cum Laude.

Fabian's research focused on collaborative autonomous driving, safe reinforcement learning, and adaptive control barrier functions. These areas reflect his strengths in robotics, hardware and his willingness to engage with advanced mathematical concepts. The author may be reached at fabian.hernandezone@gmail.com.