Safe and Robust Imitation Learning for Drone Navigation in Cluttered Construction Environments

Yun Seok Gwon[1] **and Heung Jin Oh**[1]

[1]Bowen School of Construction, Purdue University, USA

ygwon@purdue.edu, oh367@purdue.edu

Abstract -

Autonomous navigation in cluttered construction environments is challenging due to unstructured obstacles, GPS-denied conditions, and operational constraints on onboard computation. Optimization-based planners such as Covariant Hamiltonian Optimization for Motion Planning (CHOMP) offer obstacle-avoidance capabilities but are often unsuitable for high-frequency control due to their computational cost and sensitivity to local minima. This study investigates a safety-regularized imitation learning framework that integrates residual policy learning, a learned safety critic providing probabilistic risk estimates, and Conditional Value-at-Risk (CVaR) regularization. Through systematic evaluation in a procedurally generated PyBullet construction environment featuring dense scaffolding and debris, we show that safety-oriented loss shaping alone is insufficient to mitigate covariate shift in behavioral cloning, resulting in low task success despite conservative trajectories. In contrast, incorporating Dataset Aggregation (DAgger) enables the learned policy to acquire effective recovery behaviors, improving navigation success to 80% while maintaining safety margins comparable to the expert planner. The primary contribution of this work is an empirical demonstration that dataset aggregation, rather than safety regularization alone, is critical for robust imitation learning in cluttered construction environments. The proposed framework provides a foundation for distilling optimization-based planners into lightweight, reactive policies suitable for simulated aerial navigation in construction-like environments, while real-world deployment remains future work.

Keywords -

Imitation Learning; Drone Navigation; Control Barrier Functions; Robust Control; PyBullet; Robotics

1 Introduction

Construction sites are dynamic and hazardous environments for robotic deployment, characterized by narrow corridors, hanging scaffolding, and scattered debris such as pallets, pipes, and steel beams. These conditions limit the effectiveness of standard GPS-based waypoint following or map-heavy global planners for aerial robots tasked with site inspection, progress monitoring, or surveillance [1]. Therefore, navigation in such environments often requires a reactive controller that supports real-time and online obstacle avoidance control.

While optimization-based motion planners such as CHOMP [2] can generate feasible trajectories in cluttered environments, their computation costs can limit use in real-time control. Running such optimizers online at the control frequency required for quadrotors (often > 50 Hz) is difficult on standard embedded hardware [3]. Furthermore, these planners may generate trajectories with high jerk due to gradient-based updates on non-convex cost landscapes, potentially imposing mechanical loads on the airframe.

Given the computational load of optimization based motion planners, Imitation Learning (IL) has been explored as a practical alternative method for distilling the capabilities of a heavy planner into a lightweight neural policy. In this work, IL is used to learn a reactive policy from expert demonstrations [4]. However, common IL approaches, such as standard Behavioral Cloning (BC), are known to suffer from two flaws. First, it is susceptible to covariate shift, where small errors accumulate, which can drive the robot into unvisited states where the policy is undefined [5]. Second, BC mimics the expert directly; if the expert minimizes time by grazing obstacles (as CHOMP often does), the student policy inherits this behavior.

To address these limitations, this project implemented a safety-regularized imitation learning for cluttered construction environments. The approach combines residual policy learning, a learned approximation to a control barrier function, and risk-sensitive loss terms. The implementation focused on the following components:

1. A meta-residual policy learning architecture that combines a classical nominal controller with a learned neural correction to improve tracking performance and obstacle avoidance in cluttered environments.
2. A latent safety critic inspired by control barrier functions that estimates proximity to unsafe regions directly from raw LiDAR observations.
3. A risk-sensitive training objective incorporating Conditional Value-at-Risk (CVaR) to emphasize rare but

critical failure cases arising from complex obstacle geometries.

4. A systematic evaluation in a procedurally generated PyBullet construction environment to empirically demonstrate dataset aggregation, rather than safety regularization alone, is critical for robust imitation learning in highly cluttered spaces.

2 Research Background

2.1 Autonomous Navigation in Cluttered Construction Environments

Construction sites are challenging for autonomous navigation due to dense structural elements, narrow passages, unstructured debris, and frequent layout changes. These environments can be GPS-denied or unreliable, complicating GPS-dependent navigation [6]. UAVs are increasingly used for construction tasks such as site inspection, image- or video-based 3D mapping, and performance monitoring, motivating robust navigation in complex settings [7].

UAV navigation is commonly categorized into map-based and map-less (reactive) strategies. Map-based methods (e.g., SLAM and occupancy mapping) build an environment representation to support planning, whereas map-less methods react directly to sensor observations [8]. Prior work has extensively studied sampling-based and optimization-driven planners for collision-free trajectory generation, but these methods often assume stable environment representations and can be computationally demanding for onboard use on resource-constrained platforms [9].

In cluttered, dynamic settings, maintaining accurate and up-to-date maps is difficult due to layout changes, sensing artifacts, and moving obstacles, which can reduce the effectiveness of map-heavy global planning. This motivates reactive control approaches that map observations to actions without expensive online optimization. Learning-based navigation, including imitation learning, provides one such approach by distilling expert behavior into reactive policies [10]. However, robustness and safety remain challenging under distribution shift and rare failure cases, motivating imitation learning methods that incorporate safety constraints and distribution-alignment mechanisms [11].

2.2 Imitation Learning and Meta-Learning

IL has evolved beyond simple BC. While BC [12] treats control as a supervised regression problem, it lacks long-horizon stability due to covariate shift. DAgger [5] addresses this by querying the expert on student-induced states, a strategy employed in this project for data aggregation. Recent work has explored meta-learning formulations (e.g., MAML [13]) to improve adaptation across navigation tasks. While these methods enable rapid parameter updates, they primarily adjust the behavior of an underlying planner rather than altering the optimization landscape.

In highly cluttered environments, local minima often affect planner performance. Residual Policy Learning (RPL) [14] provides a complementary approach by enabling the learned component to modify the effective vector field guiding the trajectory. Generative Adversarial Imitation Learning (GAIL) [15] seeks to align occupancy measures between expert and learner but may produce unsafe behavior when expert demonstrations approach obstacle boundaries. Diffusion-based policies [16] have demonstrated multimodal action modeling; however, their iterative denoising procedure incurs computational overhead, presenting challenges for high-frequency aerial control loops.

2.3 Trajectory Optimization: CHOMP

The expert demonstrator used in this study is based on CHOMP (Covariant Hamiltonian Optimization for Motion Planning) [2]. While Model Predictive Control (MPC) [17] and sampling-based methods like RRT* are standard in local planning, MPC requires solving convex optimization problems at every timestep, which can be computationally demanding. RRT* produces probabilistically complete but often non-differentiable paths.

CHOMP treats trajectory generation as a functional gradient descent problem. It handles cost maps (ESDFs) and produces smooth trajectories, though it is prone to local minima and requires gradient computations. Our approach uses CHOMP as a *nominal* prior to guide the student policy training.

2.4 Safe Control and Barrier Functions

Control Barrier Functions (CBFs) [18] provide a method to ensure forward invariance of a safe set. If a valid barrier function $h(x)$ exists, safety can be enforced by constraining the control input u:

$$\dot{h}(x) + \alpha(h(x)) \geq 0 \tag{1}$$

Defining an analytical $h(x)$ for a raw point cloud is difficult in unstructured environments. Recent work [19] learns CBFs from data. This project implements a latent safety critic inspired by CBFs to estimate the safety margin directly from LiDAR features.

3 Preliminaries

This section defines the robot dynamics, environmental representation, and safety formulation used in the implementation.

3.1 Quadrotor Dynamics

The quadrotor is modeled as a kinematic system controlled by velocity commands. Let state $s_t \in \mathcal{S} \subset \mathbb{R}^n$ include the robot's position $p_t \in \mathbb{R}^3$, velocity $v_t \in \mathbb{R}^3$, and sensor observations o_t (LiDAR scans). The control input $u_t \in \mathcal{A} \subset \mathbb{R}^3$ specifies the target velocity vector.

$$p_{t+1} = p_t + u_t \Delta t \tag{2}$$

This kinematic model is used for high-level trajectory planning, assuming a lower-level flight controller handles the rotor dynamics.

3.2 Euclidean Signed Distance Fields (ESDF)

The expert planner relies on an implicit surface representation of the environment. An ESDF, denoted as $\phi(x)$, maps a point $x \in \mathbb{R}^3$ to the distance to the nearest obstacle boundary ∂O.

$$\phi(x) = \begin{cases} d(x, \partial O) & \text{if } x \notin O \\ -d(x, \partial O) & \text{if } x \in O \end{cases} \tag{3}$$

This representation allows for gradient computation $\nabla \phi(x)$, which is utilized by the CHOMP optimizer. In the simulation, this is computed from the LiDAR point cloud.

3.3 Control Barrier Functions (CBF)

A Control Barrier Function $h(s)$ is a scalar function defined over the state space such that the set $C = \{s \in \mathcal{S} : h(s) \geq 0\}$ represents the safe region. The boundary $\partial C = \{s : h(s) = 0\}$ represents the limit of safety. For a control system $\dot{s} = f(s, u)$, the safety condition is maintained if the control input u satisfies:

$$\frac{\partial h}{\partial s} f(s, u) \geq -\alpha h(s) \tag{4}$$

where $\alpha > 0$ determines how aggressively the system is allowed to approach the boundary. In this study, this condition is incorporated as a penalty term in the loss function rather than being solved online via QP.

3.4 Conditional Value-at-Risk (CVaR)

To address rare collision scenarios, CVaR is employed. For a loss distribution Z and confidence level $\alpha \in (0, 1)$, CVaR is defined as the expected loss in the worst α-percentile of cases:

$$\text{CVaR}_\alpha(Z) = \mathbb{E}[Z \mid Z \geq \text{VaR}_\alpha(Z)] \tag{5}$$

where VaR_α is the Value-at-Risk threshold. This term is intended to robustify the policy against outlier scenarios.

4 Methodology

4.1 System Architecture

The system consists of three main components: a global planner (providing the goal), a local expert planner (CHOMP), and the learning-based student policy. The student policy processes raw LiDAR data and state information to generate velocity commands.

4.2 Residual Policy Learning

We employed Residual Policy Learning (RPL) rather than mapping observations directly to absolute actions. The final control command u_t is given by:

$$u_t = v_{curr} + \pi_\theta(f(s_t)) + u^z_{PID} \tag{6}$$

where v_{curr} is the current velocity, π_θ predicts a delta-velocity correction (Δv) in the xy-plane, and u^z_{PID} is a fixed PID controller for altitude maintenance.

4.2.1 Feature Extraction

Raw LiDAR scans are converted into a local occupancy grid and processed into an Euclidean Signed Distance Field (ESDF) patch (size 7×7) centered on the drone. The policy network input is a concatenation of:

- The flattened ESDF patch (49 dimensions).
- Current velocity vector (3 dimensions).
- Goal delta vector (3 dimensions).

4.2.2 Network Architecture

The policy network π_θ is a Multi-Layer Perceptron (MLP) with two hidden layers of 256 units each, using LeakyReLU activations (slope 0.2).

4.3 Learned Safety Critic

A neural network $r_\phi(s)$ (Safety Critic) was trained to approximate the risk of a state. The ground truth risk R_{gt} is computed during expert data collection using an exponential decay of the minimum LiDAR clearance d_{min}:

$$R_{gt}(s) = \exp(-\gamma \cdot d_{min}) \tag{7}$$

where $\gamma = 2.0$. This signal is bounded in $[0, 1]$. The critic network mirrors the policy architecture but outputs a single scalar value through a Sigmoid activation, trained via Mean Squared Error (MSE):

$$\mathcal{L}_{critic} = \|r_\phi(s) - R_{gt}(s)\|^2 \tag{8}$$

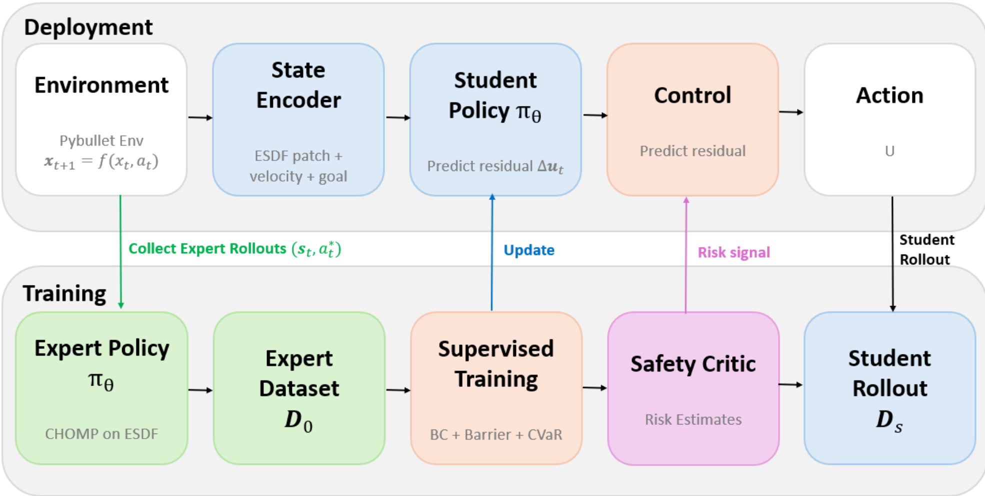

Figure 1. System architecture. The student policy learns a residual correction to the current velocity, guided by the expert's demonstrations and regulated by the Safety Critic.

4.4 Loss Function

The training process utilized a Safety-Regularized loss function, combining behavioral cloning with barrier-based regularization and risk sensitivity.

$$\begin{aligned}\mathcal{L}_{total} = &\underbrace{\|\pi_\theta(s) - \Delta u^*_{target}\|^2}_{\text{BC}} \\ &+ \lambda_1 \underbrace{\mathbb{E}[B(s,u)]}_{\text{Barrier Mean}} \\ &+ \lambda_2 \underbrace{\text{CVaR}_\alpha(B(s,u))}_{\text{Robust Safety}}\end{aligned} \tag{9}$$

where the Barrier Cost $B(s,u)$ is defined as:

$$B(s,u) = r_\phi(s) \cdot \|u_{final}\|^2 \tag{10}$$

This term penalizes high kinetic energy ($\|u\|^2$) when the estimated risk $r_\phi(s)$ is high. The $\text{CVaR}_{0.1}$ term calculates the mean of the top 10% of barrier costs in the batch. Parameters were set to $\lambda_1 = 0.01$ and $\lambda_2 = 0.05$ for the experiments.

4.5 Training Pipeline: DAgger

The DAgger algorithm was utilized to address distribution shift. The pipeline proceeds as follows:

1. **Expert Collection:** Collect initial trajectories using the CHOMP teacher.
2. **Pre-training:** Train π_θ and r_ϕ on expert data.
3. **Iterative Rollout:** Execute the student policy π_θ in the environment to collect new observations $s_{student}$.
4. **Expert Labeling:** Query the teacher for the optimal action u^* at these new states.
5. **Aggregation & Retraining:** Add the new (s, u^*) pairs to the dataset and retrain.

5 Experimental Setup

5.1 Simulation Environment

A custom simulation environment was developed using the PyBullet physics engine. The environment, "Construction Site," spans a $22m \times 22m$ area.

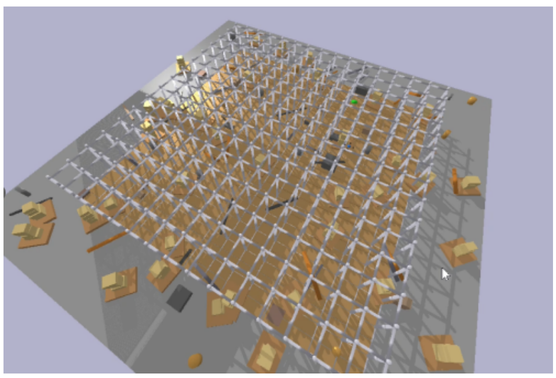

Figure 2. The PyBullet simulation environment featuring procedurally generated scaffolding and debris.

The scene generation is procedural to explicitly model the structural variability found across different construction scenarios. It consists of:

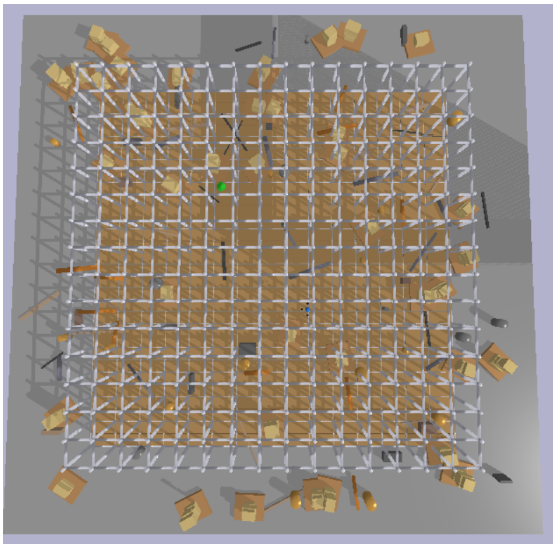

Figure 3. Top-down view of a generated episode. Red indicates the goal, blue indicates the start. Note the density of the scaffolding grid.

- **Scaffolding Grid:** A 15×15 grid of bays ($1m \times 1m$) constructed from vertical uprights and horizontal ledger pipes (radius $0.06m$).
- **Debris Scattering:** 40 wooden pallets ($0.6m \times 0.5m$) and 80 random debris items (steel beams, cylinders, pipes) scattered across the floor.
- **Spawn Logic:** The drone and goal are often spawned inside the scaffolding structure or atop debris piles.

5.2 Sensors and Robot Model

The robot is modeled as a quadrotor with a $0.18m$ radius, equipped with a simulated LiDAR sensor:

- **Range:** 10.0 meters.
- **Resolution:** 72 azimuth rays $\times$ 5 elevation layers (Total 360 rays).
- **Update Rate:** 60 Hz.
- **Feature Space:** Raw point cloud rasterized into a 7×7 ESDF patch.

Camera inputs were not used to reduce computational overhead, relying on geometric depth data. These procedurally generated scenes are intended to represent a type of cluttered construction-like layouts rather than all construction scenarios, and the present evaluation is limited to short-horizon navigation in dense obstacle fields.

5.3 Expert Demonstrator

The expert policy is based on CHOMP, utilizing the instantaneous ESDF to compute a velocity gradient minimizing:

$$\mathcal{U}(traj) = w_{obs}\mathcal{U}_{obs} + w_{smooth}\mathcal{U}_{smooth} + w_{att}\mathcal{U}_{att} \tag{11}$$

Parameters were tuned to $k_{att} = 1.5$ (goal attraction) and $k_{rep} = 0.8$ (obstacle repulsion).

6 Results

6.1 Quantitative Comparison

Five comparative baselines were evaluated to systematically analyze the performance of Imitation Learning variants against the optimization-based expert. The baselines are:

1. **Teacher (CHOMP):** The optimization-based expert.
2. **Vanilla BC:** Behavioral Cloning on expert data.
3. **BC + CVaR:** BC with risk-sensitive loss.
4. **BC + CVaR + CBF:** Full loss function on static expert data.
5. **BC + CVaR + CBF + DAgger:** The full method with iterative dataset aggregation.

Five comparative baselines were evaluated to systematically analyze the performance of Imitation Learning variants against the optimization-based expert.

Table 1 summarizes the results averaged over 15 evaluation episodes.

6.2 Observation of "Drift" Failure Mode (Methods 2-4)

Our initial hypothesis was that a robust loss function (CVaR + CBF) might enable training a safe policy from expert data alone. However, methods trained purely on static expert datasets (Methods 2, 3, and 4) yielded success rates below 15%. While these methods produced smooth trajectories (Jerk $< 8m/s^3$) and maintained high safety margins (Clearance $> 0.3m$), they consistently failed to reach the goal. Flight logs indicate these failures were due to covariate shift. The expert trajectories lie on a specific manifold; when the student drifted due to approximation error, it entered unvisited states. Lacking data in these regions, the policy output defaulted to passive behaviors, causing the drone to freeze or drift until timeout. This observation suggests that loss shaping alone was insufficient to resolve the distribution shift problem in this environment.

6.3 Impact of DAgger (Method 5)

Implementing DAgger (Dataset Aggregation) improved the success rate to 80%. By iteratively collecting data where the student operates and the expert corrects, the policy learned recovery behaviors. Notably, the jerk metric increased significantly (from 7.8 to $30.2m/s^3$). This

Table 1. Failure Analysis and Final Success (Averaged over 15 Episodes)

Method	Success	Jerk (m/s^3)	Min Clearance
Teacher (CHOMP)	100%	42.4	0.06 m
Vanilla BC	13.3%	4.3	0.55 m
BC + CVaR	6.7%	4.8	0.69 m
BC + CVaR + CBF	6.7%	7.8	0.34 m
Ours (with DAgger)	**80.0%**	**30.2**	**0.07 m**

suggests that the ”smoothness” observed in the static-data policies was a result of passivity. The DAgger-trained policy exhibited more aggressive control behavior (closer to the Teacher’s 42.4 jerk) to actively correct errors and navigate the clutter.

6.4 Flight Smoothness

The Teacher policy exhibited high jerk ($\approx 42.1 m/s^3$) due to the discrete nature of the ESDF gradient updates and the repulsion forces in the potential field. The policy with DAgger reduced jerk to $\approx 30.2 m/s^3$. This reduction indicates that the neural network effectively filtered some of the high-frequency optimization noise from the expert while maintaining the trajectory intent.

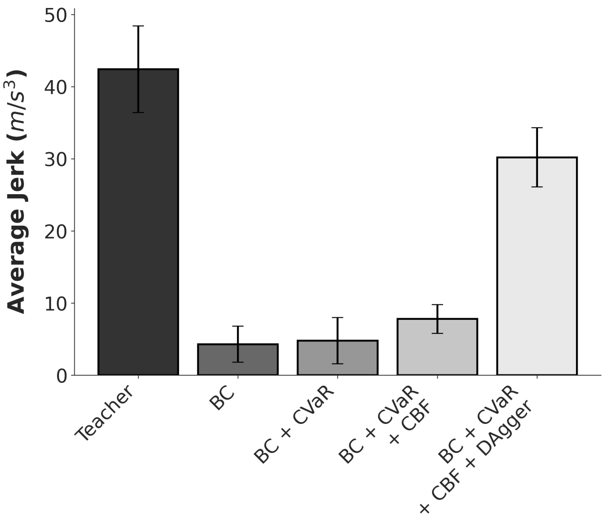

Figure 4. Comparison of average jerk. Lower values indicate smoother flight.

6.5 Safety Margins

The minimum clearance distance to obstacles was tracked during flight. The Teacher policy occasionally violated the safety margin, potentially due to local minima. The policy maintained a comparable clearance ($0.07m$ vs $0.06m$). The inclusion of the CVaR term was intended to penalize proximity to obstacles, though in the DAgger-trained model, the clearance remained tight due to the complexity of the environment.

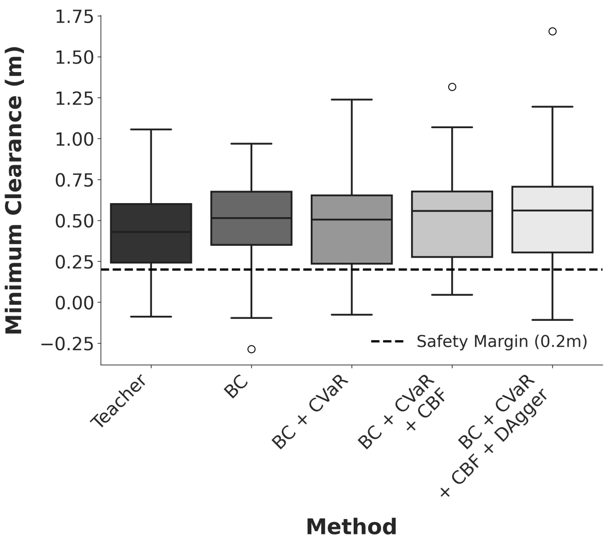

Figure 5. Distribution of minimum clearance distances.

7 Discussion

The quantitative evaluation highlights the challenges inherent in learning from static expert demonstrations. Despite incorporating safety-oriented regularization, methods trained without dataset aggregation consistently failed to achieve reliable goal-reaching performance. These failures are attributable to covariate shift: the expert trajectories occupy a narrow region of the state space, whereas deviations during execution place the learner in states for which no corresponding supervisory labels exist. Under these conditions, the policy exhibited unreliable corrective behavior.

Incorporating DAgger mitigated this issue by explicitly collecting supervisory signals in learner-induced states. This procedure enabled the policy to acquire effective recovery strategies and resulted in an improvement in success rate. The observed increase in jerk relative to static-training methods reflects a shift from conservative, lowmobility behavior toward active correction, aligning more closely with the expert's operational characteristics. This suggests a trade-off where robustness requires active correction, which inherently increases control effort. The residual-learning formulation integrated the nominal controller with the learned correction. This structure allowed the system to default to the nominal controller when the residual was near zero, though the drift issues in static training suggest the residual component is critical for navigation in complex clutter.

7.1 Scalability

The results have implications for deployment on embedded hardware. The constant-time inference $O(1)$ of the neural policy contrasts with the scaling complexity of CHOMP. This efficiency is relevant for MAVs with limited onboard compute. Additionally, the Latent CBF operates on raw depth features rather than visual textures, which visual-domain dependence in future sim-to-real transfer to physical hardware using depth cameras.

7.2 Limitations

The current method relies on a static world assumption. The ESDF is computed instantaneously, ignoring the velocity of dynamic obstacles. Extending the method to dynamic environments would require a temporal component. Additionally, the LCBF provides a probabilistic estimate rather than a formal control-theoretic guarantee of safety. The present study is limited to simulation in PyBullet and does not include real-world UAV experiments, so the results should be interpreted as controlled empirical evidence rather than deployment-ready validation.

8 Conclusion

This paper presented a safety-regularized imitation learning framework for autonomous navigation in cluttered construction environments. By combining a residual policy architecture with a learned approximation of a control barrier function and a risk-sensitive objective, the method integrates safety considerations directly into policy learning for reactive aerial navigation. Experimental results demonstrate that, although safety-aware loss terms improve clearance behavior, they are insufficient to resolve covariate shift when trained solely on static expert demonstrations. Incorporating dataset aggregation through DAgger significantly improved task success by enabling the policy to learn recovery behaviors in learner-induced states. These findings highlight the importance of distribution-alignment mechanisms in imitation learning for risk-aware robotic navigation in cluttered environments.

Beyond the simulated evaluation presented in this study, several directions are particularly relevant for construction-oriented deployment. Future work will extend the environment complexity to include a broader range of construction clutter types, such as partially assembled structural elements, temporary formwork, and moving equipment. Additional sensing degradations common to construction sites, including dust, occlusion, sparse geometry, and sensor noise, will be incorporated to better reflect real-world operating conditions. Integrating prior information from Building Information Models (BIM) as structural or semantic priors may further improve navigation robustness and enable tighter coupling with construction workflows such as progress monitoring and safety inspection. Together, these extensions aim to bridge the gap between learning-based navigation research and practical deployment of aerial robots in active construction environments.

References

[1] Aiyu Zhu, Tianhong Dai, Gangyan Xu, Pieter Pauwels, Bauke De Vries, and Meng Fang. Deep reinforcement learning for real-time assembly planning in robot-based prefabricated construction. *IEEE Transactions on Automation Science and Engineering*, 20(3):1515–1526, 2023. doi:10.1109/TASE.2023.3236805.

[2] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, 2009. doi:10.1109/ROBOT.2009.5152817.

[3] Daniel Mellinger, Nathan Michael, and Vijay Kumar. *Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors*, pages 361–373. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-28572-1. doi:10.1007/978-3-642-28572-1_25. URL `https://doi.org/10.1007/978-3-642-28572-1_25`.

[4] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL `https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf`.

[5] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learn-

ing and structured prediction. *CoRR*, abs/1011.0686, 2010. URL http://arxiv.org/abs/1011.0686.

[6] Yingxiu Chang, Yongqiang Cheng, Umar Manzoor, and John Murray. A review of uav autonomous navigation in gps-denied environments. *Robotics and Autonomous Systems*, 170:104533, 2023. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2023.104533. URL https://www.sciencedirect.com/science/article/pii/S0921889023001720.

[7] Mohammad Nahangi, Adam Heins, Brenda McCabe, and Angela Schoellig. Automated localization of uavs in gps-denied indoor construction environments using fiducial markers. In Jochen Teizer, editor, *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 88–94, Taipei, Taiwan, July 2018. International Association for Automation and Robotics in Construction (IAARC). ISBN 978-3-00-060855-1. doi:10.22260/ISARC2018/0012.

[8] Dipraj Debnath, Fernando Vanegas, Juan Sandino, Ahmad Faizul Hawary, and Felipe Gonzalez. A review of uav path-planning algorithms and obstacle avoidance methods for remote sensing applications. *Remote Sensing*, 16(21), 2024. ISSN 2072-4292. URL https://www.mdpi.com/2072-4292/16/21/4019.

[9] Gopi Gugan and Anwar Haque. Path planning for autonomous drones: Challenges and future directions. *Drones*, 7(3), 2023. ISSN 2504-446X. doi:10.3390/drones7030169. URL https://www.mdpi.com/2504-446X/7/3/169.

[10] Caleb Harris, Youngjun Choi, and Dimitri Mavris. Imitation learning for uas navigation in cluttered environments. 01 2021. doi:10.2514/6.2021-0452.

[11] Chenxin Yang. Review of ai-based uav navigation in gps-denied environments. In *Proceedings of the 2025 2nd International Conference on Electrical Engineering and Intelligent Control (EEIC 2025)*, pages 583–596. Atlantis Press, 2025. ISBN 978-94-6463-864-6. doi:10.2991/978-94-6463-864-6_51. URL https://doi.org/10.2991/978-94-6463-864-6_51.

[12] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4950–4957. AAAI Press, 2018. ISBN 9780999241127.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017. URL https://arxiv.org/abs/1703.03400.

[14] Tom Silver, Kelsey R. Allen, Joshua B. Tenenbaum, and Leslie Pack Kaelbling. Residual policy learning. *ArXiv*, abs/1812.06298, 2018. URL https://api.semanticscholar.org/CorpusID:56327855.

[15] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4572–4580, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[16] Kostas Bekris, Kris Hauser, Sylvia Herbert, Jingjin Yu, Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Rob. Res.*, 44(10–11): 1684–1704, September 2025. ISSN 0278-3649. doi:10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

[17] Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989. ISSN 0005-1098. doi:https://doi.org/10.1016/0005-1098(89)90002-2. URL https://www.sciencedirect.com/science/article/pii/0005109889900022.

[18] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431, 2019. doi:10.23919/ECC.2019.8796030.

[19] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V. Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3717–3724, 2020. doi:10.1109/CDC42340.2020.9303785.