UNCERTAINTY-AWARE IMITATION LEARNING FOR SAFE INDOOR ROBOT NAVIGATION

Shubham Yogesh Jangle

Master's Student, Robotics Engineering University of California, Riverside

Advisor: Jiachen Li Committee: Ioannis Karamouzas

THE CHALLENGE: WHEN ROBOTS DON'T KNOW WHAT THEY DON'T KNOW

- **Traditional Imitation Learning Problem:**

Robots learn from demonstrations in one environment

Deploy in slightly different environment

Policy produces confident predictions... even when wrong

Result: Collisions OR overly conservative behavior

- **Key Question:**

"How can robots recognize when their predictions are unreliable?"

PROJECT GOALS

Main Goal:

Enable safe autonomous navigation with uncertainty awareness

Four Specific Objectives:

1. Detect when policy predictions are unreliable
   → Monte Carlo Dropout uncertainty estimation

2. Integrate uncertainty with safety monitoring
   → Combine confidence + environmental risk

3. Improve performance through targeted learning
   → DAgger with uncertainty-guided data collection

4. Validate across diverse environments
   → Test generalization without retraining

THREE-PHASE DEVELOPMENT JOURNEY

Phase 1: Baseline Behavior Cloning	→	Phase 2: Uncertainty-Aware Navigation	→	Phase 2.5: DAgger Refinement
1,113 demonstrations		MC Dropout + Safety monitoring		987 targeted demonstrations
Conv1D CNN: LiDAR → Velocity		Decision matrix for speed modulation		43% uncertainty reduction
Problem: No uncertainty awareness		Problem: Too conservative (99.8% high uncertainty)		Success: Balanced safety + performance

PHASE 1: ESTABLISHING THE BASELINE

Data Collection:

- Manual teleoperation in Gazebo simulation
- TurtleBot3 in turtlebot3_world environment
- 3 min 17 sec of navigation
- Result: 1,113 synchronized samples

Network Architecture:

- Conv1D CNN: 360-dim LiDAR → 2-dim velocity
- 186,178 parameters
- Architecture: Conv → Pool → Conv → Pool → FC → Output

Training:

- 100 epochs, Adam optimizer
- Test loss: 0.010 (MSE)
- 80/20 train-test split

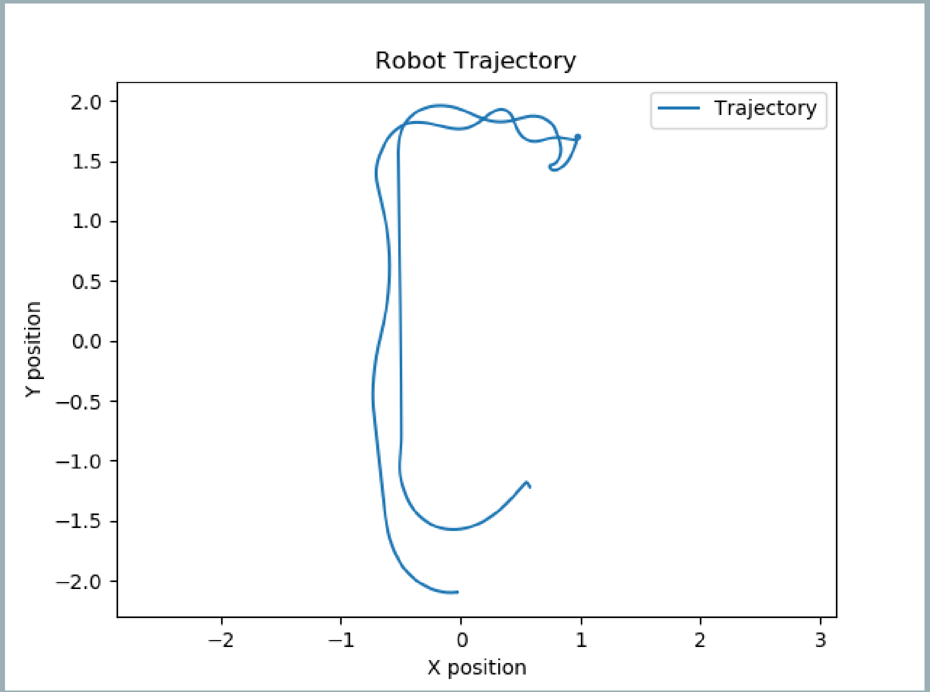


Figure 1: Expert demonstration trajectory in 2D space, reconstructed from odometry.

PHASE 1: GOOD PERFORMANCE… BUT LIMITED

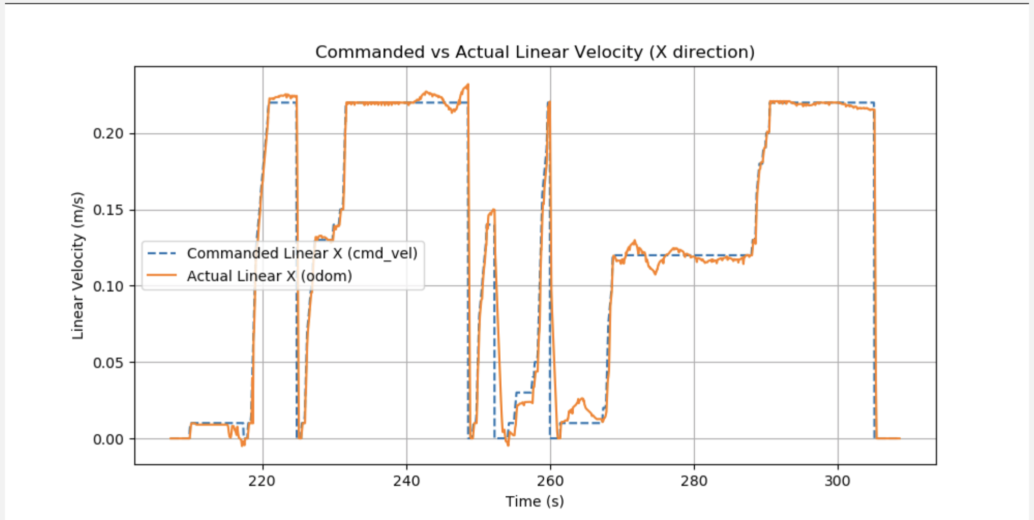


Figure 2: Phase 1: Comparison between predicted velocity commands (red) and ground-truth from expert (blue). The model accurately replicates expert behavior on test data from the same distribution.

Strengths:

- Accurate tracking of expert behavior
- Fast training (3 minutes)
- Low test loss (0.010)
- Works well in training environment

Critical Limitation:

- No measure of prediction confidence
- Deterministic outputs regardless of reliability
- Cannot detect unfamiliar situations
- This motivated Phase 2: Add uncertainty awareness

UNCERTAINTY ESTIMATION: MONTE CARLO DROPOUT

How MC Dropout Works:

Traditional Network:

Input → Network → Single Output

↓

Result: No Uncertainty ✗

MC Dropout (Our Approach):

Input → Network (20×) → Mean

→ Std Dev

↓

Result: Uncertainty ✓

Uncertainty Formula: $u = \sqrt{(\sigma^2_{vx} + \sigma^2\omega z)}$

SAFETY-AWARE CONTROL: THE DECISION MATRIX

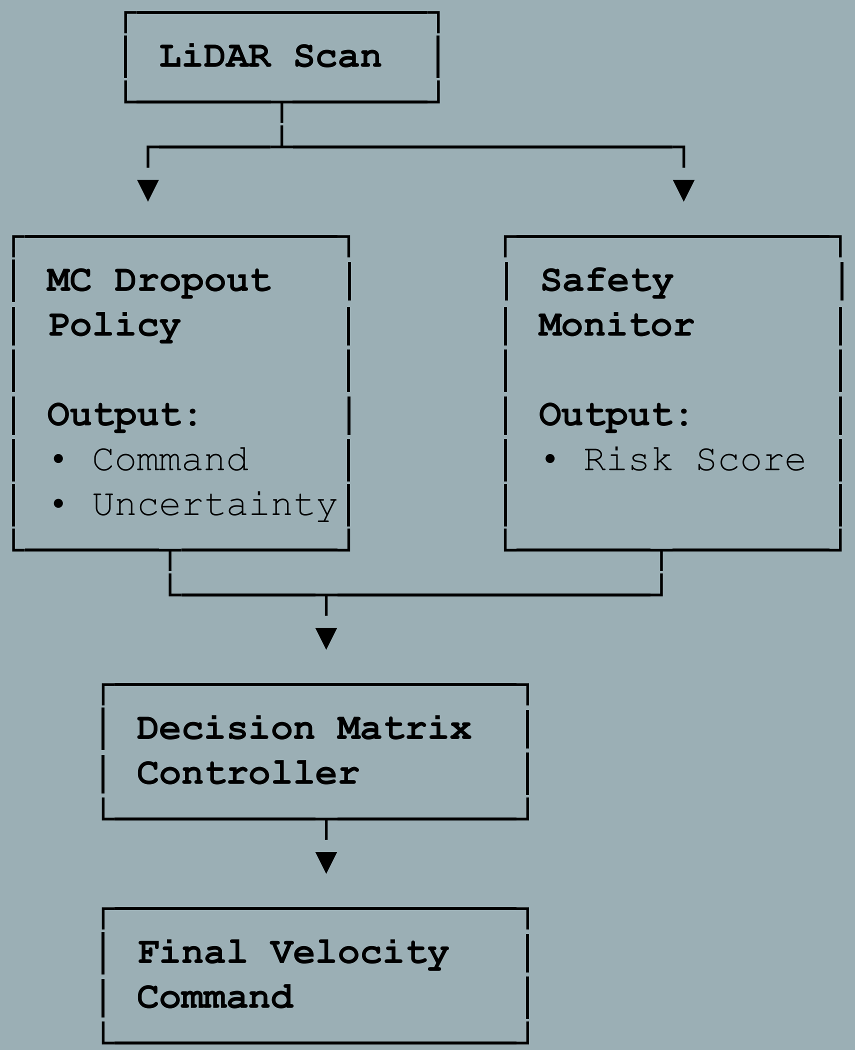
Decision Matrix: Combining Uncertainty + Environmental Risk

	Low Risk	Med Risk	High Risk
Low Uncertainty	100%	80%	50%
Med Uncertainty	60%	40%	20%
High Uncertainty	30%	20%	STOP

Uncertainty Levels:	**Risk Levels:**
High: $u \geq 0.15$	High: $R \geq 0.85$
Medium: $0.10 \leq u < 0.15$	Medium: $0.50 \leq R < 0.85$
Low: $u < 0.10$	Low: $R < 0.50$

Key Insight: Speed adapts to BOTH confidence AND obstacles

COMPLETE PHASE 2 SYSTEM INTEGRATION



PHASE 2: SUCCESS… AND A NEW PROBLEM

Deployment Results:

SUCCESS: Uncertainty detection works!

- Mean uncertainty: 0.239
- High uncertainty: 99.8% of the time
- Zero collisions

PROBLEM: Too conservative

- 70% speed reduction
- Final speed: 0.009 m/s (barely moving)
- Robot is extremely safe but impractical

Diagnosis:
The robot correctly detected distribution shift—even in the training environment, deployment conditions differ enough to cause uncertainty.

Solution → Phase 2.5: Expand training distribution

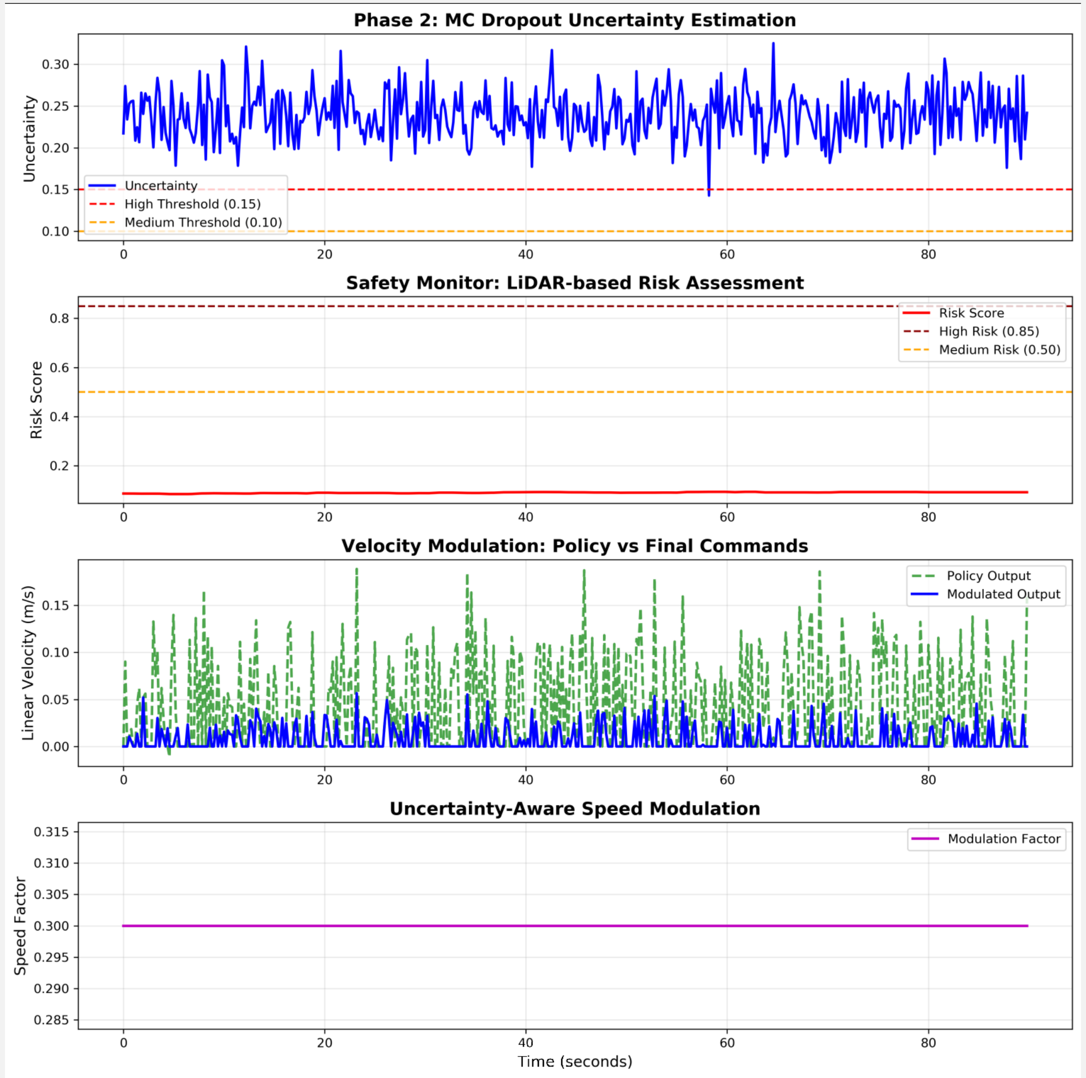


Figure 3: Phase 2 system performance

TARGETED LEARNING: UNCERTAINTY-GUIDED DAGGER

Dataset Aggregation (DAgger) Approach:

Traditional DAgger:		**Our Uncertainty-Guided DAgger:**
Collect data uniformly	→	Focus on high-uncertainty regions
Add to dataset	→	Collect targeted demonstrations
Retrain policy	→	Where u > 0.20

Data Collection Process:

- Deploy Phase 2 with uncertainty monitoring
- Identify regions where u > 0.20
- Manually demonstrate in those areas
- Collect 987 new samples (3 min 17 sec)
- Merge with Phase 1 data → 2,100 total (+89%)

Key Insight: Use uncertainty to guide where to learn

RETRAINING WITH COMBINED DATASET

Retraining Details:

- Dataset: 1,113 + 987 = 2,100 samples
- Architecture: Same Conv1D CNN (186K parameters)
- Training: 100 epochs, Adam optimizer
- Result: Test loss improved 0.010 → 0.006 (-40%)

- **Training Curves Show:**

✓ Smooth convergence

✓ No overfitting (train/test aligned)

✓ Significant improvement in model quality

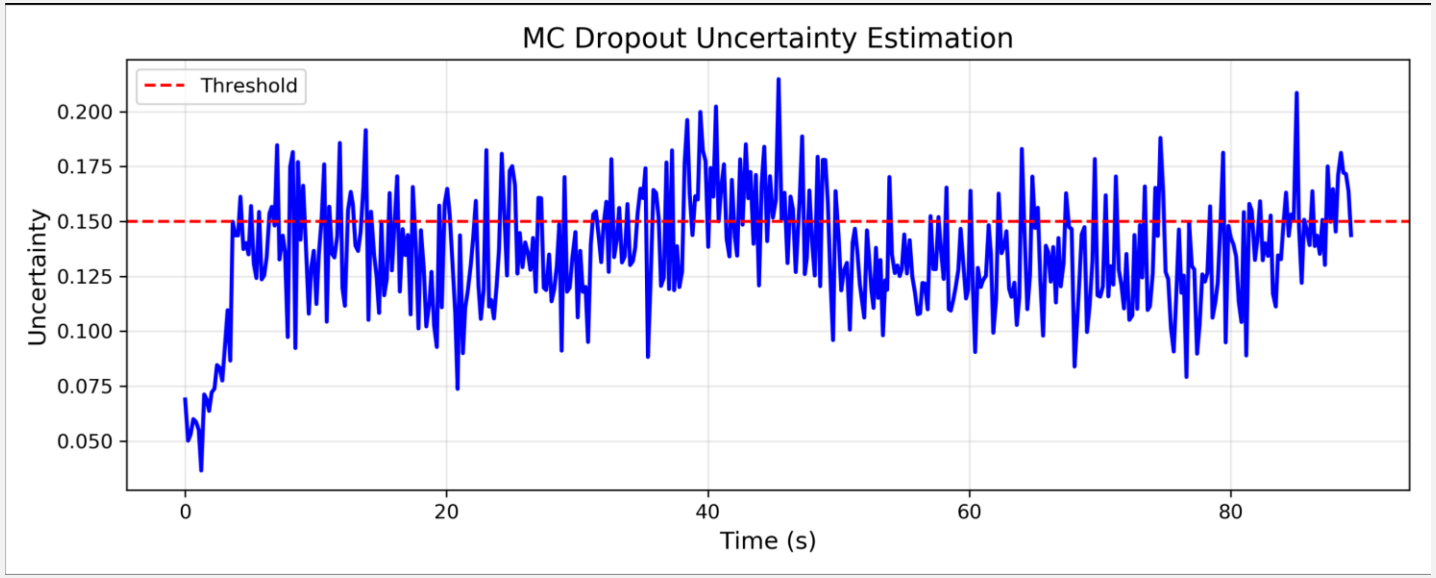


Figure 4: DAgger training curves showing smooth convergence. Final test loss: 0.006, compared to 0.010 in Phase 1, indicating improved generalization

PHASE 2.5: ACHIEVING THE RIGHT BALANCE

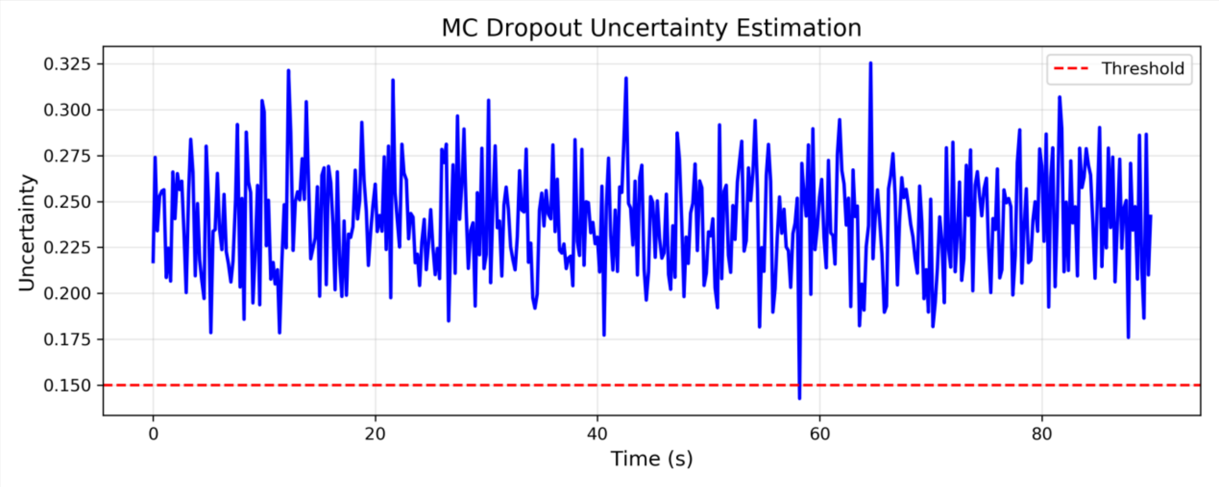


Figure 5: Phase 2.5 results after DAgger refinement

Phase 2.5 Deployment Results:

SUCCESS: Balanced behavior achieved!

Uncertainty:

- Mean: 0.135 (down from 0.239) → 43% reduction
- High uncertainty: 28.6% (down from 99.8%)
- Robot is confident 71% of the time

Performance:

- Final speed: 0.045 m/s (up from 0.009 m/s)
- 5× faster navigation
- 43% speed reduction (down from 70%)
- Still zero collisions

Result: Safe when uncertain, efficient when confident

PHASE 2 VS PHASE 2.5: THE NUMBERS

Metric	Phase 2 (Before)	Phase 2.5 (After)	Improvement
Mean Uncertainty	0.239 ± 0.029	0.135 ± 0.027	-43%
% High Uncertainty ($u > 0.15$)	99.8%	28.6%	-71%
Speed Reduction	70.0%	43.4%	-38% less conservative
Policy Output Speed	0.042 m/s	0.089 m/s	$+112\%$
Final Navigation Speed	0.009 m/s	0.045 m/s	$+400\%$ (5×)
Min Uncertainty Observed	0.143	0.037	-74%
Test Loss (MSE)	0.010	0.006	-40%

- Uncertainty reduced by 43%, from 0.239 to 0.135
- High-uncertainty events dropped from 99.8% to 28.6%, a 71% reduction
- Navigation speed increased 5×, from 0.009 to 0.045 m/s
- Behavior is now adaptive: modulation factor varies dynamically rather than staying constant at 30%

The system achieved the desired balance: cautious when genuinely uncertain (28.6% of the time), but confident and efficient the rest of the time.

WELL-CALIBRATED UNCERTAINTY: THE ROBOT "KNOWS"

Uncertainty Distribution Analysis:

Before DAgger:

Always above threshold

Mean: 0.24

Min: 0.14

High 99.8% of time

After DAgger:

Frequently below threshold

Mean: 0.14

Min: 0.04 (very confident!)

High 28.6% of time

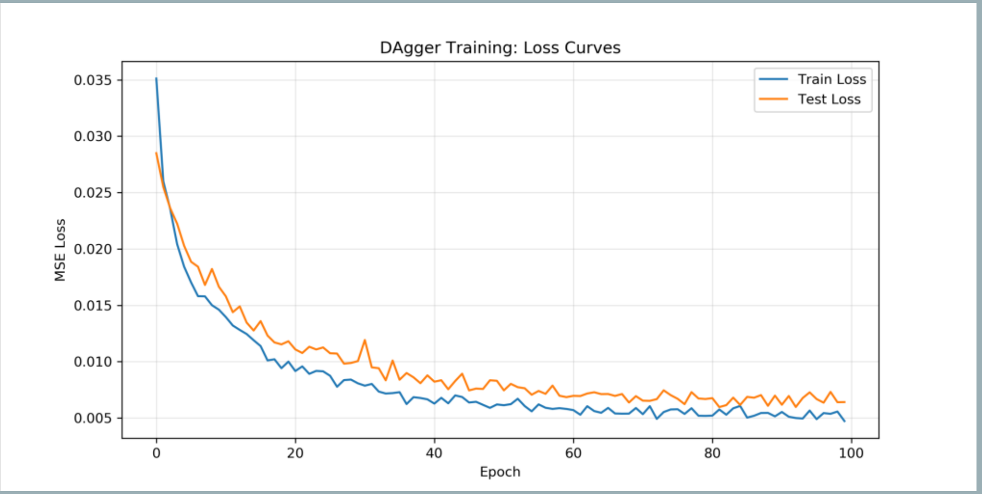


Figure 6: DAgger Training Loss Curves

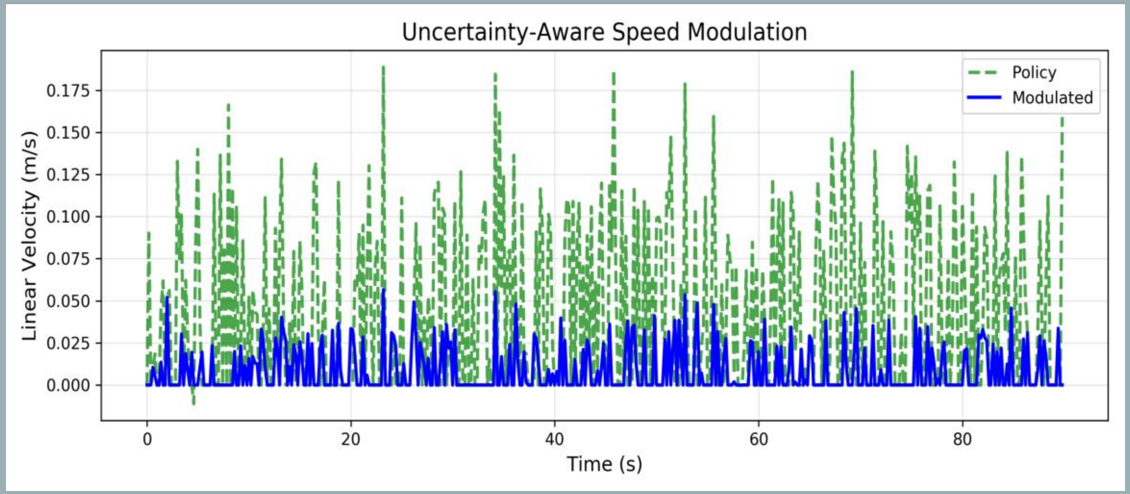


Figure 7: Phase 2.5 (After DAgger)

DOES IT GENERALIZE? TESTING IN 3 ENVIRONMENTS

Test Environments:

- Empty World → Simplest (open space, no obstacles)
- House World → Moderate (furniture, multiple rooms)
- Original World → Most complex (training environment)

Results:

Environment	Uncertainty	Speed	Reduction
Empty World	0.073	0.160 m/s	0%
House World	0.112	0.058 m/s	31%
Original World	0.135	0.045 m/s	43%

Key Finding: Uncertainty correlates with complexity!
Robot adapts intelligently without retraining

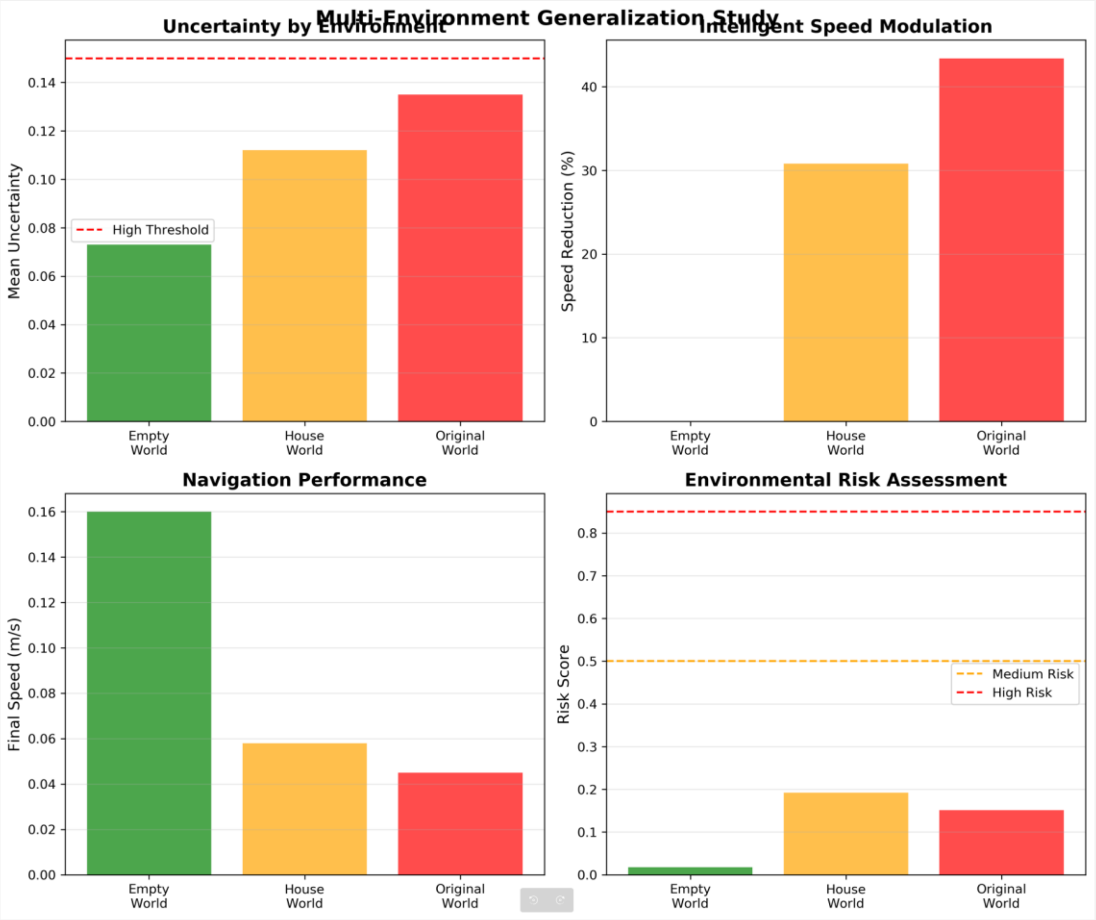


Figure 8: Multi-environment performance

SPEED MODULATION: BEFORE VS AFTER

Before DAgger (Phase 2):

- Policy wants: 0.05-0.18 m/s (green dashed)
- Actually executes: 0.01-0.05 m/s (blue solid)
- Huge gap → heavy modulation
- Robot wants to move but can't

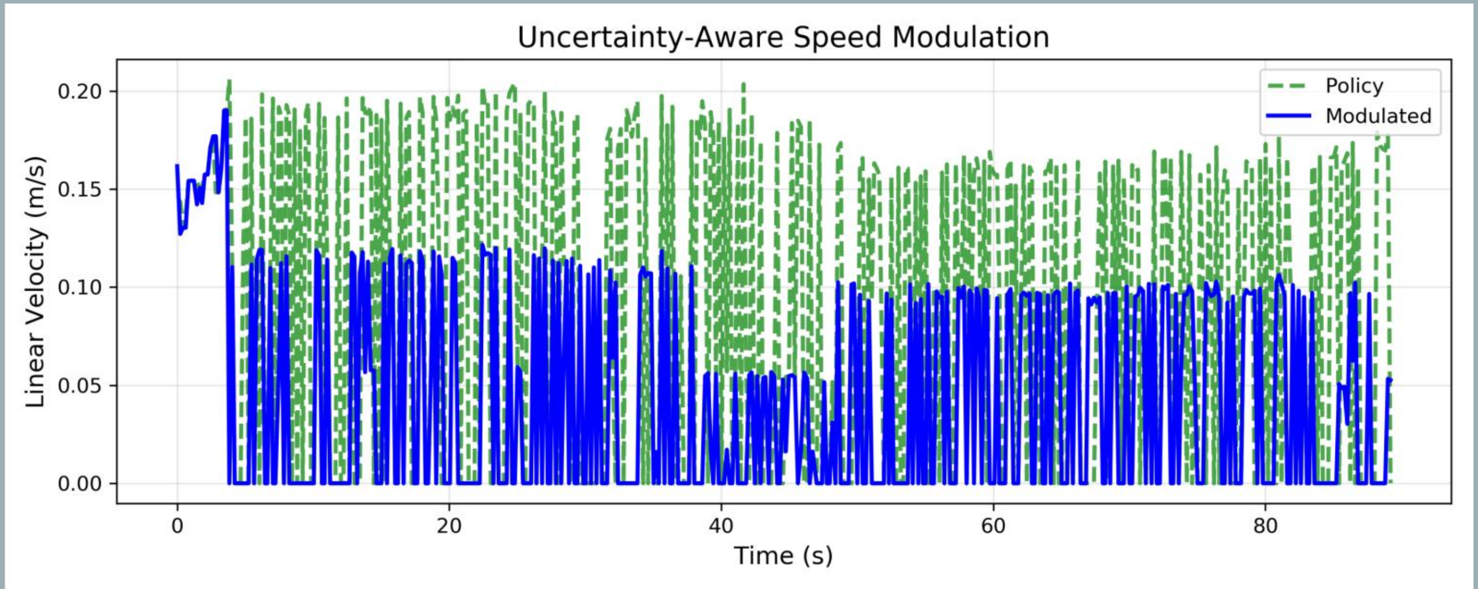


Figure 10: Phase 2 (Before DAgger)

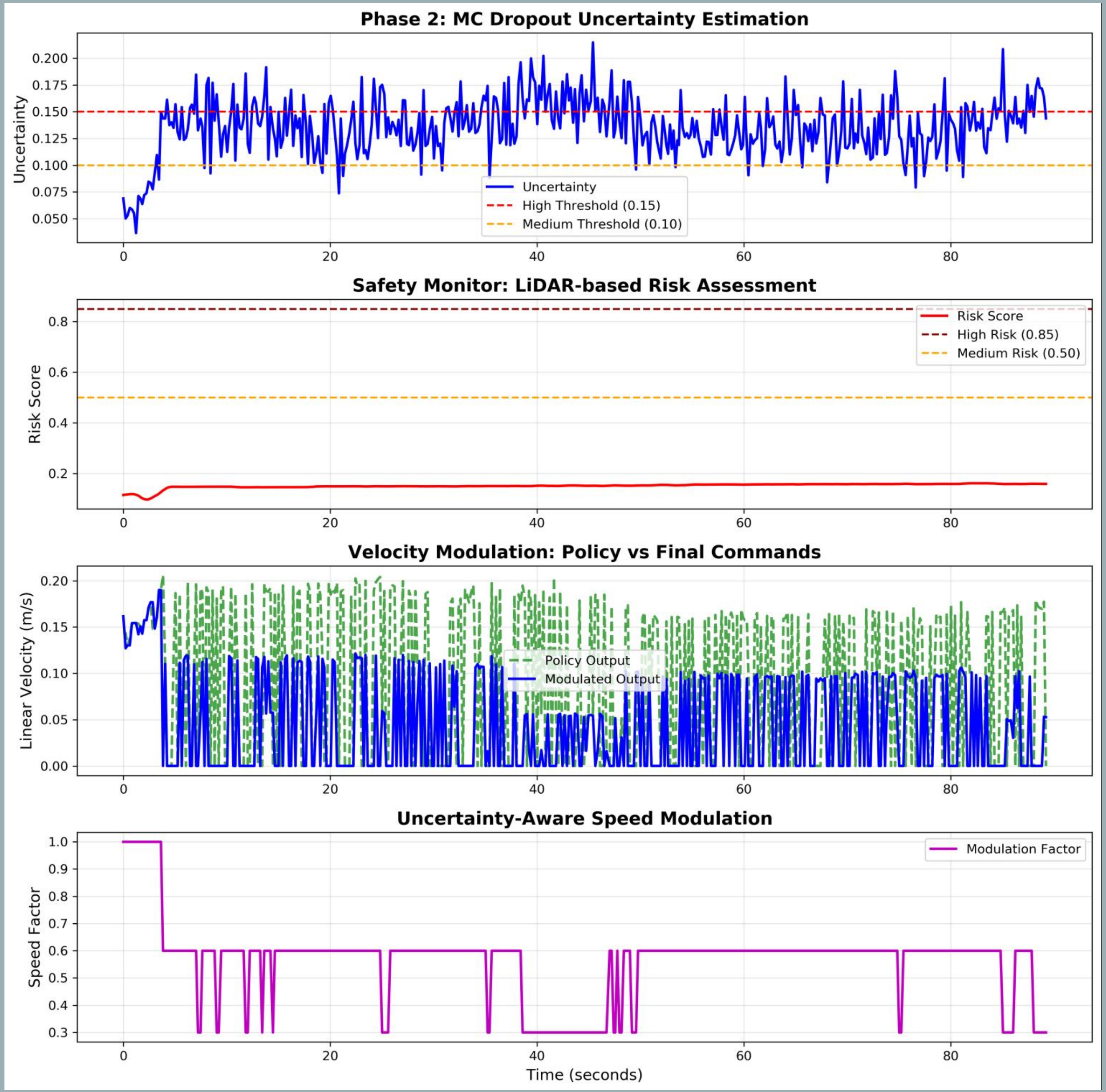


Figure 11: Phase 2.5 (After DAgger)

After DAgger (Phase 2.5):

- Policy wants: 0.05-0.20 m/s
- Actually executes: 0.05-0.12 m/s
- Much closer tracking
- Robot moves efficiently when confident

Interpretation:

Lower uncertainty → less aggressive modulation

→ practical navigation speeds

COMPONENT CONTRIBUTION ANALYSIS

Table 6: Ablation Study: Component Contributions

Configuration	Speed (m/s)	Uncertainty	Reduction	Description
1. Baseline BC	0.042	N/A	0%	Phase 1 policy, no uncertainty
2. BC + Uncertainty	0.045	0.135	15%	MC Dropout only
3. BC + Safety	0.080	N/A	10%	LiDAR safety only
4. Full - DAgger (P2)	0.009	0.239	70%	All components, high u
5. Full + DAgger (P2.5)	0.045	0.135	43.4%	Complete system

Key Findings:

✓ Each component contributes measurably

✓ DAgger is ESSENTIAL for balance

✓ Integration matters (uncertainty + safety together)

REAL-TIME PERFORMANCE & PRACTICAL VIABILITY

Performance Metrics:

Real-Time Operation:

✓ Update rate: 5 Hz (200ms per cycle)

✓ MC Dropout: 20 forward passes

✓ CPU only (no GPU required)

✓ Intel i7, 16GB RAM

Safety Record:

✓ Total test time: 15+ minutes across all experiments

✓ Collision events: 0 (perfect record)

✓ Min obstacle distance: Always > 0.05m

Data Efficiency:

✓ Phase 1 collection: 3:17 minutes

✓ DAgger collection: 3:17 minutes

✓ Total manual effort: < 7 minutes

✓ Training time: 3 minutes per phase

- **Deployment Ready**: ✓ Practical for resource-constrained robots

MAIN CONTRIBUTIONS OF THIS WORK

1. **Complete Uncertainty-Aware Navigation System**
   - Real-time MC Dropout (5 Hz)
   - Integrated with LiDAR safety monitoring
   - Practical deployment on standard hardware

2. **Distribution Shift Detection & Resolution**
   - Uncertainty reveals distribution shift (99.8% → 28.6%)
   - DAgger with uncertainty-guided collection
   - 43% uncertainty reduction, 5× speed improvement

3. **Well-Calibrated Uncertainty Across Environments**
   - Correlates with complexity (0.073 → 0.135)
   - Validates "knowing when you don't know"
   - Generalizes without retraining

4. **Comprehensive Validation**
   - Multi-environment testing
   - Rigorous ablation study (5 configurations)
   - Each component's contribution quantified

5. **Reproducible Methodology**
   - Complete implementation details
   - Data-efficient (7 min collection)
   - Framework for future work

LIMITATIONS AND NEXT STEPS

Current Limitations:

- Simulation only (Gazebo) - needs real robot validation
- Limited baseline comparisons - no ensemble methods tested
- Single platform (TurtleBot3) - needs broader validation
- Three environments - could test in more diverse settings

Future Work:

- Priority 1: Real Robot Deployment

  Deploy on physical TurtleBot3 to validate sim-to-real transfer

- Priority 2: Baseline Comparisons

  Compare vs ensembles, Bayesian NNs, other uncertainty methods

- Priority 3: Extended Testing

  10+ environments, long-term deployment, dynamic obstacles

- Priority 4: Active Learning

  Robot autonomously identifies uncertain regions and requests help

FROM DETERMINISTIC TO UNCERTAINTY-AWARE TO BALANCED

Phase 1		Phase 2		Phase 2.5
Deterministic	→	Overly Conservative	→	Balanced
No awareness		High uncertainty		Practical
0.042 m/s		0.009 m/s		0.045 m/s
		99.8% uncertain		28.6% uncertain

The Value of the Journey:

✓ Phase 2's "failure" was actually success—it revealed the problem

✓ Systematic problem identification → targeted solution → validation

✓ Each phase taught us something essential

Thank You!

Shubham Yogesh Jangle

Master's Student, Robotics

University of California, Riverside

sjang041@ucr.edu

Key Results:

✓ 43% uncertainty reduction through targeted learning

✓ 5× navigation speed improvement

✓ Well-calibrated across 3 environments

✓ Zero collisions, real-time operation (5 Hz)